

## PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2002-049575

(43)Date of publication of application : 15.02.2002

(51)Int.Cl.

G06F 13/14

G06F 3/06

G06F 12/00

G06F 12/16

G06F 13/00

(21)Application number : 2000-233291

(71)Applicant : HITACHI LTD

(22)Date of filing : 01.08.2000

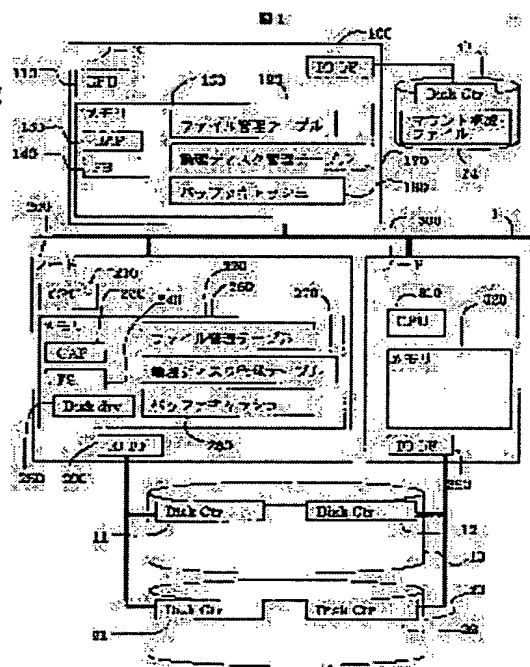
(72)Inventor : ITO AKIHIRO  
UTSUNOMIYA NAOKI  
SONODA KOJI  
KUMAZAKI HIROYUKI

## (54) FILE SYSTEM

## (57)Abstract:

PROBLEM TO BE SOLVED: To provide a file system which can shorten the time needed for IO path switching and conceal IO path switching processing from general users.

SOLUTION: In this system which defines file IDs for every files, a file server FS refers to a file management table and finds a logical disk ID for accessing a file when a user application UAP makes a request to access while specifying the file ID of the file. Further, the file server refers to a logical disk management table to find the IO path corresponding to the logical disk ID and uses the IO path to access a physical disk unit. If trouble occurs to the IO path of an in-operation system, the logical disk management tables of all nodes are rewritten to switch the IO path.



## LEGAL STATUS

[Date of request for examination]

18.07.2003

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision]

(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号

特開2002-49575

(P2002-49575A)

(43)公開日 平成14年2月15日(2002.2.15)

(51)Int.Cl. <sup>7</sup>	識別記号	F I	テーマコード(参考)
G 0 6 F 13/14	3 1 0	G 0 6 F 13/14	3 1 0 H 5 B 0 1 4
3/06	3 0 4	3/06	3 0 4 B 5 B 0 1 8
12/00	5 1 4	12/00	5 1 4 E 5 B 0 6 5
12/16	3 1 0	12/16	3 1 0 A 5 B 0 8 2
13/00	3 0 1	13/00	3 0 1 P 5 B 0 8 3

審査請求 未請求 請求項の数20 O L (全 26 頁)

(21)出願番号 特願2000-233291(P2000-233291)

(22)出願日 平成12年8月1日(2000.8.1)

(71)出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72)発明者 伊藤 昭博

神奈川県川崎市麻生区王禅寺1099番地 株

式会社日立製作所システム開発研究所内

(72)発明者 宇都宮 直樹

神奈川県川崎市麻生区王禅寺1099番地 株

式会社日立製作所システム開発研究所内

(74)代理人 100078134

弁理士 武 顕次郎

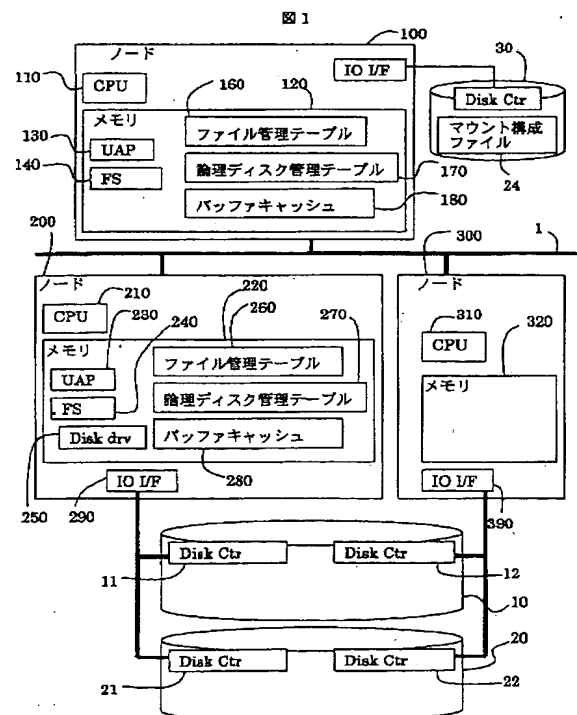
最終頁に続く

(54)【発明の名称】 ファイルシステム

(57)【要約】

【課題】 I Oパス切り替えのために要する時間を短縮し、一般ユーザからI Oパス切り替え処理を隠蔽することのできるファイルシステム。

【解決手段】 ファイル毎にファイルIDが定義されているシステムにおいて、ユーザアプリケーションUAPからのファイルIDを指定したアクセス要求に対して、ファイルサーバFSはファイル管理テーブルを参照し、そのファイルをアクセスするための論理ディスクIDを求める。ファイルサーバは、さらに、論理ディスク管理テーブルを参照し、論理ディスクIDに対応するI Oパスを求め、そのI Oパスを使って物理ディスク装置にアクセスする。運用系のI Oパスに障害発生時、全ノードの論理ディスク管理テーブルを書き換えることによってI Oパスの切り替えを行う。



## 【特許請求の範囲】

【請求項1】 ファイル毎にファイルIDが定義されており、複数の物理ディスク装置に分散管理されたファイルの処理を行う1または複数のファイルサーバを有するファイルシステムにおいて、ファイルID及び該ファイルIDに対応するファイルが格納されている論理ディスクの論理ディスクIDを含むファイル管理テーブルと、論理ディスクID及び前記論理ディスクに対応する1つ以上の物理ディスク装置にアクセスするための1つ以上のIOパスを含む論理ディスク管理テーブルとを備え、ユーザからのファイルIDを指定したファイルへのアクセス要求を受信したファイルサーバは、ファイル管理テーブルを参照し、前記ファイルIDから前記ファイルが格納されている論理ディスクの論理ディスクIDを決定し、論理ディスク管理テーブルを参照して前記論理ディスクIDから前記論理ディスクに対応する物理ディスク装置にアクセスするためのIOパスを決定し、決定したIOパスを使用して物理ディスク装置にアクセスすることを特徴とするファイルシステム。

【請求項2】 ネットワークに接続されたそれぞれの内部にファイルサーバが構成された複数のノードと、複数のノードの少なくとも2つのノードに共通に接続された物理ディスク装置とを備え、ファイル毎にファイルIDが定義されており、前記複数の物理ディスク装置に分散管理されたファイルの処理を行うファイルシステムにおいて、複数のノードのそれぞれは、ファイルID及び前記ファイルIDに対応するファイルが格納されている論理ディスクの論理ディスクIDを含むファイル管理テーブルと、論理ディスクID及び前記論理ディスクに対応する1つ以上の物理ディスク装置にアクセスするための1つ以上のIOパスを含む論理ディスク管理テーブルとを備え、ユーザからのファイルIDを指定したファイルへのアクセス要求を受信したファイルサーバは、ファイル管理テーブルを参照し、前記ファイルIDから前記ファイルが格納されている論理ディスクの論理ディスクIDを決定し、論理ディスク管理テーブルを参照して前記論理ディスクIDから前記論理ディスクに対応する物理ディスク装置にアクセスするためのIOパスを決定し、決定したIOパスを使用して物理ディスク装置にアクセスすることを特徴とするファイルシステム。

【請求項3】 前記IOパスを特定する情報は、ノード番号、IOインターフェイス番号及びディスクコントローラ番号からなることを特徴とする請求項2記載のファイルシステム。

【請求項4】 前記ファイルIDから決定した論理ディスクIDに対応する物理ディスク装置が、他のノードであるリモートノードに接続されている場合、自ノードのファイルサーバは、前記リモートノードにアクセス要求を送信し、前記アクセス要求を受信した前記リモートノードのファイルサーバが前記物理ディスク装置に格納さ

れた該当ファイルにアクセスすることを特徴とする請求項3記載のファイルシステム。

【請求項5】 ネットワークに接続されたそれぞれの内部にファイルサーバが構成された複数のノードと、複数のノードの少なくとも2つのノードに共通に接続された物理ディスク装置とを備え、ファイル毎にファイルIDが定義されており、前記複数の物理ディスク装置に分散管理されたファイルの処理を行うファイルシステムにおいて、前記物理ディスク装置の少なくとも1つは、1つのマウントポイントに対して物理ディスク装置にアクセスするための1つ以上のIOパスを対応づける情報を1つのエントリに含むマウント構成ファイルを格納しており、システム立ち上げ時、前記マウント構成ファイルを格納するディスク装置が接続されたノードのファイルサーバは、前記マウント構成ファイルを読み出し、前記マウント構成ファイルの1つのエントリに記載された1つ以上のIOパスに対して1つの論理ディスクIDを自動設定し、前記論理ディスクIDと前記IOパスとの対応関係を論理ディスク管理テーブルに登録し、他の全てのノードのファイルサーバと通信を行うことによって、前記論理ディスク管理テーブルの内容を全てのノードの論理ディスク管理テーブルに複写し、前記マウント構成ファイルによって前記IOパスに対応づけられたマウントポイントに前記論理ディスクIDに対応する論理ディスクをマウントし、複数のノードのそれぞれは、ファイルID及び前記ファイルIDに対応するファイルが格納されている論理ディスクの論理ディスクIDを含むファイル管理テーブルと、論理ディスクID及び前記論理ディスクに対応する1つ以上の物理ディスク装置にアクセスするための1つ以上のIOパスを含む論理ディスク管理テーブルとを備え、ユーザからのファイルIDを指定したファイルへのアクセス要求を受信したファイルサーバは、ファイル管理テーブルを参照し、前記ファイルIDから前記ファイルが格納されている論理ディスクの論理ディスクIDを決定し、論理ディスク管理テーブルを参照して前記論理ディスクIDから前記論理ディスクに対応する物理ディスク装置にアクセスするためのIOパスを決定し、決定したIOパスを使用して物理ディスク装置にアクセスすることを特徴とするファイルシステム。

【請求項6】 前記マウント構成ファイルは、IOパス毎に前記IOパスが使用できるか否かを登録する使用可否情報を含み、論理ディスク管理テーブルは、前記論理ディスク管理テーブルに登録されているIOパス毎に稼働状態を保持する状態フラグを含み、マウント処理を行うファイルサーバは、システム立ち上げ時に、前記マウント構成ファイルの1つのエントリに記載された複数のIOパスのうち、前記マウント構成ファイルの使用可否情報に「使用可」と登録されたIOパスの1つについて、論理ディスク管理テーブルの前記IOパスに対応する状態フラグに「使用中」状態と登録し、前記マウント

構成ファイルの使用可否情報に「使用可」と登録された残りの I/Oパスについて、前記論理ディスク管理テーブルの前記 I/Oパスに対応する状態フラグに「待機中」状態と登録し、前記マウント構成ファイルの使用可否情報に「使用不可」と登録された I/Oパスについて、前記論理ディスク管理テーブルの前記 I/Oパスに対応する状態フラグに「使用不可」状態と登録し、各ノードのファイルサーバは、通常運用時、前記論理ディスク管理テーブルの状態フラグが「使用中」状態となっている運用系の I/Oパスを用いて、物理ディスク装置にアクセスすることを特徴とする請求項 5 記載のファイルシステム。

【請求項 7】 前記物理ディスク装置のディスクコントローラ、前記物理ディスク装置が接続されたノードの I/Oインターフェイスなどの障害によって、運用系 I/Oパスが使用不可能になったとき、前記障害を検出したノードのファイルサーバは、前記ノードの論理ディスク管理テーブルを更新し、前記使用不可能になった I/Oパスの状態フラグを「使用不可」とし、前記使用不可能になった I/Oパスと同じ論理ディスク ID に対応付けられている I/Oパスのうち状態フラグが「待機中」である 1 つの I/Oパスの状態フラグを「使用中」として新運用系 I/Oパスとした後、全ての他のリモートノードのファイルサーバと通信を行い、前記論理ディスク管理テーブルの内容を全ノードの論理ディスク管理テーブルに複製することによって、前記物理ディスク装置へアクセスするための I/Oパスを前記使用不可能となった I/Oパスから前記新運用系 I/Oパスに切り替えることを特徴とする請求項 6 記載のファイルシステム。

【請求項 8】 前記 I/Oパスの切り替え処理の間、使用不可能となった I/Oパスに含まれるノードのファイルサーバは、使用不可能になった I/Oパスへのアクセス要求を保留し、I/Oパスの切り替え処理終了時、保留していた前記アクセス要求を新運用系 I/Oパスに含まれるノードに転送することを特徴とする請求項 7 記載のファイルシステム。

【請求項 9】 前記 I/Oパスの切り替え処理の間、使用不可能となった I/Oパスに含まれるノードにアクセス要求を発行したファイルサーバは、前記アクセス要求がタイムアウトになった場合、論理ディスク管理テーブルを参照し論理ディスク ID から I/Oパスを求め直し、新しく求め直した I/Oパスを使用して、物理ディスク装置にアクセスし直すことを特徴とする請求項 7 記載のファイルシステム。

【請求項 10】 前記複数のノードのそれぞれは、物理ディスク装置との間に転送されるデータを一時的に保持するバッファキャッシュを備え、I/Oパスの切り替え処理時、使用不可能になった I/Oパスに含まれるノードのファイルサーバと、新運用系 I/Oパスに含まれるノードのファイルサーバとが通信を行い、前記使用不可能になった I/Oパスに含まれるノードの主記憶内に存在し、物

理ディスク装置に書き戻す必要があるバッファキャッシュ及びファイル管理テーブルを前記新運用系 I/Oパスに含まれるノードに転送することを特徴とする請求項 7 記載のファイルシステム。

【請求項 11】 前記物理ディスク装置内のディスクコントローラは、ディスク領域との間で転送されるデータを一時的に保持するディスクキャッシュを備え、前記物理ディスク装置内の別のディスクコントローラが備えるディスクキャッシュに格納されたデータをディスク領域に書き戻す機能を有し、I/Oパスの切り替え処理時、前記使用不可能になった I/Oパスを使ってアクセスしていた物理ディスク装置内に設けられた前記使用不可能になった I/Oパスに含まれるディスクコントローラが備えるディスクキャッシュに格納されたデータのうち、前記物理ディスク装置に書き戻す必要のあるデータを、前記物理ディスク装置内に存在し、新運用系 I/Oパスに含まれるディスクコントローラを使用して、前記物理ディスク装置に書き戻すことを特徴とする請求項 7 記載のファイルシステム。

【請求項 12】 I/Oパスの切り替え終了時、マウント構成ファイルを格納するディスク装置が接続されたノードのファイルサーバが前記マウント構成ファイルを更新し、前記使用不可能となった I/Oパスの使用可否情報を「使用不可」に書き換えることを特徴とする請求項 7 記載のファイルシステム。

【請求項 13】 使用不可能となっていた I/Oパスが再び使用できるようになったとき、前記複数のノードのある 1 つのノードのファイルサーバが、自ノードの論理ディスク管理テーブルに登録された前記 I/Oパスの状態フラグを「使用不可」状態から「待機中」状態に更新し、前記ファイルサーバが他の全てのノードのファイルサーバと通信を行うことにより、全てのノードの論理ディスク管理テーブルに前記更新内容を複製した後、マウント構成ファイルを格納するディスク装置が接続されたノードのファイルサーバが、前記マウント構成ファイルに登録された前記 I/Oパスの使用可否情報を「使用可」に書き換えることにより、前記 I/Oパスを待機系 I/Oパスとしてシステムに復旧させることを特徴とする請求項 7 記載のファイルシステム。

【請求項 14】 物理ディスク装置が接続されたノードに障害が発生したとき、前記ノードの障害を検出した他のノードのファイルサーバは、自ノードの論理ディスク管理テーブルを検索し、障害発生ノード番号から障害発生 I/Oパス及び前記障害発生 I/Oパスと同じ論理ディスク ID に対応付けられている I/Oパスのうち状態フラグが「待機中」である I/Oパスの 1 つを新運用系 I/Oパスとして求め、この新運用系 I/Oパスに含まれるノードのファイルサーバに I/Oパスの切り替え処理を行うように要求し、前記要求を受けた前記ファイルサーバは、自ノードの論理ディスク管理テーブルを更新し、前記障害発

生I/Oパスの状態フラグを「使用不可」とし、前記新運用系I/Oパスの状態フラグを「使用中」とした後、他の全てのノードのファイルサーバと通信を行い、前記論理ディスク管理テーブルの内容を全ノードの論理ディスク管理テーブルに複写することによって、前記物理ディスク装置へアクセスするためのI/Oパスを前記障害発生I/Oパスから前記新運用系I/Oパスに切り替えることを特徴とする請求項6記載のファイルシステム。

【請求項15】 I/Oパスの切り替え処理の間、前記障害発生I/Oパスに含まれるノードにアクセス要求を発行したファイルサーバは、前記アクセス要求がタイムアウトになった場合、論理ディスク管理テーブルを参照し論理ディスクIDからI/Oパスを求め直し、新しく求め直したI/Oパスを使用して、物理ディスク装置にアクセスし直すことを特徴とする請求項14記載のファイルシステム。

【請求項16】 前記物理ディスク装置が接続されているノードは、自ノードの状態にかかわらず自ノードが備えるメモリ内のデータを読み出し、読み出したデータを他のノードに転送する機能を持ったハードウェアを有し、I/Oパスの切り替え処理時、前記ハードウェアを用いて、前記障害発生I/Oパスに含まれるノードの主記憶内に存在し、物理ディスク装置に書き戻す必要があるバッファキャッシュ及びファイル管理テーブルを前記新運用系I/Oパスに含まれるノードに転送することを特徴とする請求項14記載のファイルシステム。

【請求項17】 前記物理ディスク装置内のディスクコントローラは、ディスク領域との間で転送されるデータを一時的に保持するディスクキャッシュを備え、前記物理ディスク装置内の別のディスクコントローラが備えるディスクキャッシュに格納されたデータをディスク領域に書き戻す機能を有し、I/Oパスの切り替え処理時、前記障害発生I/Oパスを使ってアクセスしていた物理ディスク装置内に設けられた前記障害発生I/Oパスに含まれるディスクコントローラが備えるディスクキャッシュに格納されたデータのうち、前記物理ディスク装置に書き戻す必要のあるデータを、前記物理ディスク装置内に存在し、新運用系I/Oパスに含まれるディスクコントローラを使用して、前記物理ディスク装置に書き戻すことを特徴とする請求項14記載のファイルシステム。

【請求項18】 I/Oパスの切り替え終了時、マウント構成ファイルを格納するディスク装置が接続されたノードのファイルサーバが前記マウント構成ファイルを更新し、使用できなくなった運用系I/Oパスの使用可否情報を「使用不可」に書き換えることを特徴とする請求項14記載のファイルシステム。

【請求項19】 前記マウント構成ファイルは、I/Oパス毎に前記I/Oパスが使用できるか否かを登録する使用可否情報を含み、前記論理ディスク管理テーブルは、該論理ディスク管理テーブルに登録されているI/Oパス毎

に稼働状態を保持する状態フラグを含み、前記マウント処理を行うファイルサーバは、システム立ち上げ時に、前記マウント構成ファイルの使用可否情報に「使用可」と登録されたI/Oパスについて、論理ディスク管理テーブルの前記I/Oパスに対応する状態フラグに「使用中」状態と登録し、前記マウント構成ファイルの使用可否情報に「使用不可」と登録されたI/Oパスについて、論理ディスク管理テーブルの前記I/Oパスに対応する状態フラグに「使用不可」状態と登録し、通常運用時、ファイルサーバは、前記論理ディスク管理テーブルの状態フラグが「使用中」状態のI/Oパスからアクセスされる物理ディスク装置にファイルをミラーリングすることの特徴とする請求項5記載のファイルシステム。

【請求項20】 前記使用中I/Oパスの1つに障害が発生したとき、この障害を検出したノードのファイルサーバは、自ノードの論理ディスク管理テーブルを更新し、障害が発生した前記I/Oパスの状態フラグを「使用不可」とした後、他の全てのノードのファイルサーバと通信を行い、前記論理ディスク管理テーブルの内容を全ノードの論理ディスク管理テーブルに複写し、マウント構成ファイルを格納するディスク装置が接続されたノードのファイルサーバが、前記マウント構成ファイルを更新し、前記障害が発生したI/Oパスの使用可否情報を「使用不可」に書き換えることによって、障害が発生したI/Oパスを切り放すことを特徴とする請求項19記載のファイルシステム。

#### 【発明の詳細な説明】

##### 【0001】

【発明の属する技術分野】本発明は、複数のディスク装置に分散管理されたファイルの処理を行うファイルシステムに係り、特に、1つのディスク装置へアクセスするためのI/Oパスが複数存在する場合に、I/Oパスの切り替えを制御を行って一方のパスからディスク装置へアクセスすることができるファイルシステムに関する。

##### 【0002】

【従来の技術】従来技術によるファイルシステムの1つであるUNIX（登録商標）ファイルシステムは、各ファイル毎にユニークに決まる番号（ファイルID）が定義されており、ファイルサーバがファイルIDを指定することによって、リード・ライト処理を行うファイルを特定することができる。そして、ファイルサーバは、ファイルIDとそのファイルが格納されているディスク装置にアクセスするためのI/Oパス（I/Oパスを決定する情報は、ノード番号、I/Oインターフェイス番号、装置番号などである）との対応関係をメモリ上のファイル管理テーブル（UNIXではinodeと呼ばれる）に登録して管理している。この管理方法については、例えば、(The Design of The Unix Operating System; Maurice J. Bach; p60-p72)に述べられている。

【0003】ファイルIDを指定したリード・ライトア

アクセス要求に対して、ファイルサーバは、前述のファイル管理テーブルを参照し、ファイルIDからディスク装置にアクセスするためのI/Oパス名を決定し、そのI/Oパスを用いてディスク装置にアクセスを行う。ファイル管理テーブルには、I/Oパス情報の他に、ファイルサイズやファイルの更新日付などのファイル管理情報が登録されており、このファイル管理情報は、ファイルがオープンされたとき、ディスク装置から読み出され、定期的あるいはファイルをクローズしたときに、ディスク装置に書き戻される。ユーザがファイルにアクセスするとき指定するファイル名からファイルIDへの変換は、ファイルサーバが行っている。

【0004】また、複数のディスク装置をシステムで取り扱う場合、あるディスク装置Aで管理されるディレクトリネームツリー内のいずれかのディレクトリ、例えば、Xに別のディスク装置Bで管理されるネームツリーを組み込むという操作によって、複数のディスク装置を1つのネームツリー内に見せるという方法が知られている。この方法によれば、ユーザは、ディレクトリXにアクセスすればディスク装置B内のファイルにアクセスすることができる。この方法は、マウント処理と呼ばれているものである。ファイルサーバは、起動時にある特定のディスク装置（ルートデバイス）を起点として前述したマウント処理を次々に行い、ユーザには複数のディスク装置を1つのネームツリーとして見せるようにしている。この起動時におけるマウント処理を行うためのディスク装置とネームツリー上のディレクトリ名（マウントポイント）との対応関係を記述した情報は、ルートデバイスにマウント構成ファイルとして記録されており、ファイルサーバは、起動時にこのマウント構成ファイルに記載された情報に従ってマウント処理を行う。

【0005】マウント構成ファイルには、ディスク装置を特定する情報として、そのディスク装置にアクセスするためのI/Oパスの情報が記載されている。ファイルサーバは、マウント処理の実行時に、マウント構成ファイルに記載されたI/Oパスとマウントポイントとの対応関係をメモリ上のマウント構成情報に読み込む。そして、ファイルサーバは、ユーザがファイル名を指定してファイルをオープンするとき、前述のマウント構成情報を元にファイルが格納されている物理ディスク装置にアクセスするためのI/Oパスを求め、ファイル管理テーブルを作成する。従って、システム管理者は、システムに新しいディスク装置を接続するなどしてシステムの構成を変更したとき、マウント構成ファイルを書き換えることによって、新しい構成情報を計算機システムに設定する必要がある。

【0006】一方、計算機システムの信頼性を向上させるため、異なる2つのノードを1つのディスク装置に物理的に接続し、異なる2通りのI/Oパスからディスク装置にアクセスすることができる構成にしておき、通常の

運用時に一方のI/Oパスを使用し、ノード障害が発生して使用中のI/Oパスが使用できなくなったとき、もう一方のI/Oパスを用いて別のノードからディスク装置にアクセスするようにすることによって、障害発生時においてもディスク装置の可用性（アベイラビリティ）を保つ方法が、例えば、特開平10-275090号公報等に記載されて知られている。

【0007】また、ディスク装置の信頼性を向上するために、ファイルを複数のディスクに多重化して記録する方法（ミラーリング）がよく知られている。ミラーリングを行う場合、一般に、論理ボリュームという概念が用いられる。ミラーリングは、複数の物理ディスク装置を、1つの論理ボリュームとしてユーザに見せる仕組みである。ユーザは、予め複数の物理ディスク装置の情報を登録した「論理ボリューム」を作成しておく。そして、ユーザがこの論理ボリュームに対して、物理ディスク装置と同様にアクセスすると、複数の物理ディスクへのファイルのミラーリングが行われる。論理ボリュームを使用することにより、ファイルを複数のディスク装置に分散記録するストライピングを行うことも可能となる。

【0008】

【発明が解決しようとする課題】 前述で説明した使用中のI/Oパスが使用不可能になったとき、物理ディスク装置にアクセスするためのI/Oパスを別のI/Oパスに切り替える処理を、従来のUNIXのファイルシステムに適用して動的に行おうとすると、ファイル管理テーブル及びマウント構成情報を検索し、使用できなくなったI/Oパス名を新しいI/Oパス名に書き換える操作を行う必要がある。前述のファイル管理テーブルのエントリを書き換える処理は、オープンされているファイルの個数だけ全てについて行わなければならない。この結果、従来のUNIXのファイルシステムに、前述したI/Oパスの切り替えの技術を適用した場合、ファイル管理テーブルのエントリを書き換える処理に時間がかかり、その間その物理ディスク装置にI/O処理を行うことができないという問題点を生じることになる。

【0009】また、I/Oパスに障害が発生したときに、単純にI/Oパスを切り替えるだけでは、障害発生前に物理ディスク装置にアクセスを行っていたノードが持っていたバッファキャッシュ（物理ディスク装置にリード・ライトするときにデータを一時的に蓄えておき、メモリに比べて処理速度の遅い物理ディスク装置への入出力回数を削減するためのメモリ領域）やファイル管理テーブル、及び、ディスク装置上のディスクキャッシュ（バッファキャッシュと同様の目的のために物理ディスク装置が備えるキャッシュメモリ）の内容が正常に物理ディスク装置に書き戻されず、大切なデータが消えてしまうという問題点をも生じる。しかも、これが原因でファイルシステムの整合性が異常となるため、物理ディスク装置

に冗長に記録されたファイルシステムの情報を元にファイルシステムの整合性を正常状態に戻す操作が必要となる。この操作は、ディスク装置全体をチェックする必要があるため、長い時間を要する。この結果、この間、その物理ディスク装置に対する I/O 処理を行うことはできないという問題点を生じさせてしまう。

【0010】さらに、I/Oパス切り替え後、新しい I/Oパスを用いてディスク装置にアクセスを行うので、I/Oパス切り替え後にシステムを再起動したときにマウント処理が正常に行われるようにするには、システム管理者がマウント構成ファイルを更新し、ディスク装置への新しい I/Oパスとマウントポイントとの対応関係をマウント構成ファイルに登録しなおす必要がある。また、ファイルのミラーリングを行う場合、論理ボリュームを作成する必要があるが、論理ボリュームの管理は、システム管理者に対して煩雑な作業を行わせることになる。

【0011】本発明の第1の目的は、I/Oパスの切り替え処理のために要する時間を短縮し、一般ユーザから I/Oパス切り替え処理をできるだけ隠蔽することができるファイルシステムを提供することにある。また、本発明の第2の目的は、I/Oパスの切り替え時に、バッファキャッシュやファイル管理テーブル及びディスク装置上のディスクキャッシュに保存されたデータを失うことなく I/Oパスの切り替え処理を行い、ファイルの整合性のチェックを不要とすることができるファイルシステムを提供することにある。また、本発明の第3の目的は、I/Oパスを切り替えたとき自動的にマウント構成ファイルを更新し、システム管理者の負担を軽減することのできるファイルシステムを提供することにある。さらに、本発明の第4の目的は、ユーザに論理ボリュームを意識せずにはファイルのミラーリングを行う方法を備えたファイルシステムを提供することにある。

【0012】

【課題を解決するための手段】本発明によれば前記目的は、ファイル毎にファイル ID が定義されており、複数の物理ディスク装置に分散管理されたファイルの処理を行う 1 または複数のファイルサーバを有するファイルシステムにおいて、ファイル ID 及び該ファイル ID に対応するファイルが格納されている論理ディスクの論理ディスク ID を含むファイル管理テーブルと、論理ディスク ID 及び前記論理ディスクに対応する 1 つ以上の物理ディスク装置にアクセスするための 1 つ以上の I/Oパスを含む論理ディスク管理テーブルとを備え、ユーザからのファイル ID を指定したファイルへのアクセス要求を受信したファイルサーバは、ファイル管理テーブルを参照し、前記ファイル ID から前記ファイルが格納されている論理ディスクの論理ディスク ID を決定し、論理ディスク管理テーブルを参照して前記論理ディスク ID から前記論理ディスクに対応する物理ディスク装置にアクセスするための I/Oパス（I/Oパスを決定する情報は、

ノード番号、I/Oインターフェイス番号、ディスクコントローラ番号である）を決定し、決定した I/Oパスを使用して物理ディスク装置にアクセスすることにより達成される。

【0013】前述において、論理ディスク管理テーブルは、該論理ディスク管理テーブルに登録されている I/Oパス毎に稼働状態（「使用中」、「待機中」、「使用不可」）を保持する状態フラグを含み、通常運用時、ファイルサーバは状態フラグが「使用中」状態の I/Oパス（運用系 I/Oパス）を用いて物理ディスク装置にアクセスする。前記運用系 I/Oパスの障害発生時、障害を検出したノードのファイルサーバは、前記ノードの論理ディスク管理テーブルを更新し、前記障害発生 I/Oパスの状態フラグを「使用不可」とし、状態フラグが「待機中」状態である I/Oパスの状態フラグを「使用中」として新運用系 I/Oパスとした後、全リモートノードのファイルサーバと通信を行い、前記論理ディスク管理テーブルの内容を全ノードの論理ディスク管理テーブルに複製することによって、前記物理ディスク装置にアクセスするための I/Oパスを旧運用系 I/Oパスから新運用系 I/Oパスに切り替える。

【0014】この I/Oパス切り替え処理の間、前記障害発生 I/Oパスに含まれるノードのファイルサーバは、旧運用系 I/Oパスへのアクセス要求を保留し、I/Oパス切り替え処理終了時、保留していたアクセス要求を前記新運用系 I/Oパスが含むノードに送信する。これによって、I/Oパス切り替え処理を動的に行うことが可能となり、I/Oパス切り替え時ファイル管理テーブルを検索・更新する必要をなくし、I/Oパス切り替え処理に要する時間を短縮することができる。

【0015】また、前述において、I/Oパスの切り替え処理時、使用できなくなった旧運用系 I/Oパスを使ってアクセスしていた物理ディスク装置内に設けられたディスクコントローラが有するディスクキャッシュに格納されたデータのうち、前記物理ディスク装置に書き戻す必要のあるデータを、前記物理ディスク装置内に設けられた別のディスクコントローラを使用して前記物理ディスク装置に書き戻し、前記旧運用系 I/Oパスに含まれるノードのファイルサーバと新運用系 I/Oパスに含まれるノードのファイルサーバが通信を行うことによって、前記旧運用系 I/Oパスに含まれるノードの主記憶内に存在し、前記物理ディスク装置に書き戻す必要があるバッファキャッシュ及びファイル管理テーブルを前記新運用系 I/Oパスに含まれるノードに転送する。本発明は、これによって、ディスク装置上のディスクキャッシュに存在していたデータや、バッファキャッシュや、ファイル管理テーブルが消失するのを防ぎ、ファイルシステムの整合性のチェックを不要とすることができる。

【0016】また、前述において、マウント構成ファイルは、I/Oパス毎にその I/Oパスが使用できるか否かを

登録する使用可否情報を含み、ファイルサーバは、システム起動時に前記マウント構成ファイルを読み込み、前記使用可否情報に「使用可」と記載されたＩＯパスについて、対応する論理ディスク管理テーブルの状態フラグを「使用中」または「待機中」と登録し、前記使用可否情報に「使用不可」と記載されたＩＯパスについて、対応する論理ディスク管理テーブルの状態フラグを「使用不可」と登録することにより、マウント構成ファイルに「使用可」と記載されたＩＯパスだけを使用して物理ディスク装置にアクセスをする設定を行っている。ＩＯパス切り替え・切り離し処理終了後、ファイルサーバは、前記マウント構成ファイルを更新し、使用できなくなった旧運用系ＩＯパスの使用可否情報を「使用不可」に書き換える。また、使用不可能となったＩＯパスが再び使用できるようになったとき、ファイルサーバは、マウント構成ファイルを更新し、使用可能になった前記ＩＯパスの使用可否情報を「使用可」に書き換える。このように、本発明は、ＩＯパスが切り替わったときや復旧したときのマウント構成ファイルの書き換え処理を自動化することにより、システム管理者の負担を軽減することができる。

【0017】また、本発明は、マウント構成ファイルの１つのエントリに書かれた複数のＩＯパスからアクセスされる複数のディスク装置に対して、ファイルのミラーリングを行うことができ、これにより、ユーザが論理ボリュームを使用することなくファイルのミラーリングを行うことができる。

【0018】

【発明の実施の形態】以下、本発明によるファイルシステムの実施形態を図面により詳細に説明する。

【0019】図１は本発明の第１の実施形態によるファイルシステムの構成を示すブロック図、図２はシステム内に設けられる各種のテーブルの具対的な構成例を説明する図、図３はマウント構成ファイルの具体的な構成例を説明する図である。図１～図３において、１はネットワーク、１０、２０、３０は物理ディスク装置、１１、１２、２１、２２はディスクコントローラ、２４はマウント構成ファイル、１００、２００、３００はノード、１１０、２１０、３１０はＣＰＵ、１２０、２２０、３２０はメモリ、１３０、２３０はユーザアプリケーション（ＵＡＰ）、１４０、２４０はファイルサーバ（ＦＳ）、２５０はディスクドライバ、１６０、２６０はファイル管理テーブル、１７０、２７０は論理ディスク管理テーブル、１８０、２８０はバッファキャッシュ、２９０、３９０はＩＯインターフェイスである。

【0020】本発明の第１の実施形態によるファイルシステムは、図１に示すように、超並列計算機システムを構成するノード１００、２００、３００（図１では３つのノードのみを示しているが、ノードは多数設けられる）がネットワーク１によって相互に接続されて構成さ

れている。ノード２００とノード３００とには、両ノードからアクセス可能な共用物理ディスク装置１０、２０が接続されている。物理ディスク装置１０、２０は、それらのディスク装置内に設けられたディスクコントローラ１１、１２及びノード２００内に設けられたＩＯインターフェイス２９０によってノード２００と接続されると共に、ディスクコントローラ１２、２２及びノード３００内に設けられたＩＯインターフェイス３９０によってノード３００と接続されている。ノード１００に接続されている物理ディスク装置３０は、物理ディスク装置１０、２０と比べて障害発生率が極めて低い高信頼ディスク装置である。

【0021】ノード２００は、ＣＰＵ２１０とメモリ２２０とから構成される。メモリ２２０は、ユーザアプリケーション２３０と、ファイル制御を行うファイルサーバ２４０と、ディスクＩＯ処理を行うディスクドライバ２５０と、ファイル管理テーブル２６０と、論理ディスクを定義している論理ディスク管理テーブル２７０と、バッファキャッシュ２８０とを含む。ノード１００及びノード３００は、ノード２００と同様に構成されている。

【0022】物理ディスク装置にアクセスするための入出力経路をＩＯパスと呼び、このＩＯパスは、ノード番号、ＩＯインターフェイス番号、ディスクコントローラ番号の３つの情報で決定され、ＩＯパスを決めると物理ディスク装置を一意に決めることができる。例えば、

（ノード番号、ＩＯインターフェイス番号、コントローラ番号）＝（２００，２９０，１１）というＩＯパスからは、物理ディスク装置１０にアクセスされる。以後の説明において、ＩＯパスは、前述のような形式で記載することとする。

【0023】論理ディスクは、１つ以上の物理ディスク装置を組み合わせたものとして構成される。その物理ディスクの組み合わせは、ＩＯパスを指定することによって行われる。例えば、（２００，２９０，１１）、（３００，３９０，２２）という２つのＩＯパスを組み合わせると、物理ディスク装置１０、２０を纏めた論理ディスクを構成することができる。その際、物理ディスク装置１０、２０に同一の内容を記録するようにすれば、論理ディスクをミラー化することができる。また、（２００，２９０，１１）、（３００，３９０，１２）という２つのＩＯパスを組み合わせると、これらのＩＯパスからは共に物理ディスク装置１０にアクセスされるため、物理ディスク装置１０に対応する論理ディスクが構成される。但し、この場合、物理ディスク装置１０にアクセスするためのＩＯパスが２通り存在するので、片方のＩＯパスに障害が発生した場合でも、別のＩＯパスから物理ディスク装置１０にアクセスすることができ、これによって、ディスク装置の信頼性の向上を図ることができる。説明する本発明の第１の実施形態は、論理ディスク



が1つの物理ディスク装置に対応する後者の場合を例として取り扱う。

【0024】論理ディスク管理テーブル270は、図2(b)に示すように、論理ディスクID271と、ノード番号272、276と、I/Oインターフェイス番号273、277と、ディスクコントローラ番号274、278と、状態フラグ275、279とから構成される。272~274は、論理ディスクID271に対応する物理ディスク装置にアクセスするための第1のI/Oパスを決定し、状態フラグ275には、このI/Oパスの稼働状態(「使用中」、「待機中」、「使用不可」のいずれか)が登録される。276~278は、物理ディスク装置にアクセスするための第2のI/Oパスを決定し、このI/Oパスの稼働状態が状態フラグ279に登録される。このように論理ディスク管理テーブル270には、1つの論理ディスクIDに対して2通りのI/OパスとそれぞれのI/Oパスの状態フラグを登録できるようになっている。

【0025】本発明の第1の実施形態において、前述の2つのI/Oパスからアクセスされる物理ディスク装置は同一のものであり、通常運用時は2つのI/Oパスのうち1つを使用し(状態フラグが「使用中」状態になっている)、もう一方のI/Oパスを「待機中」状態としておき、ディスクコントローラやI/Oインターフェイスの障害等の原因により、使用中のI/Oパスが使用できなくなったとき、ファイルサーバが物理ディスク装置にアクセスするためのI/Oパスを「待機中」状態のI/Oパスに切り替える。このように、論理ディスク管理テーブルは、論理ディスクIDと物理ディスク装置にアクセスするためのI/Oパスとを対応付けることによって、仮想的なディスク装置として論理ディスクを定義している。論理ディスクIDはこの論理ディスクを識別するための番号である。

【0026】また、システムを構成する各ノードが持つ論理ディスク管理テーブルの内容は常に同一となっている。例えば、図1において、ノード100が持つ論理ディスク管理テーブル170と、ノード200が持つ論理ディスク管理テーブル270と、ノード300が持つ論理ディスク管理テーブル370は常に同一の内容を有する。

【0027】ファイル管理テーブル260は、図2(a)に示すように、ファイルID261と論理ディスクID262とファイル管理情報263とにより構成される。ファイルID261には、現在オープンされているファイルのファイルIDが登録され、論理ディスクID262には、前述のファイルが格納されている論理ディスクの論理ディスクIDが登録される。ファイル管理情報263には、前述のファイルのファイルサイズや更新日付等の情報が登録される。このファイル管理テーブル260の各エントリは、ノード200上で動作するプ

ログラムがファイルをオープンする度に、物理ディスク装置上から各ファイル固有の情報として読み出される。従って、ファイル管理テーブル260のエントリは、少なくともオープンされているファイルの個数分存在する。

【0028】バッファキャッシュ280は、物理ディスク装置にアクセスを行うときにリード・ライトするデータを一時的に蓄えておき、メモリに比べて処理速度の遅い物理ディスク装置への入出力処理回数を削減するために使用される。バッファキャッシュ280は、図2

(c)に示すように、論理ディスクID281とブロック番号282とキャッシュデータ283とから構成される。キャッシュデータ283には、論理ディスクID281のブロック番号282で指定されるディスク領域のデータの内容が格納される。

【0029】高信頼な物理ディスク装置30内には、マウント構成ファイル24が格納されている。マウント構成ファイル24のエントリは、図3に示すように、システムに接続される物理ディスク装置にアクセスするためのI/Oパス名51、53と、そのI/Oパスが使用可能か否かを示す使用可否情報52、54と、前述の物理ディスク装置に対応する論理ディスクをマウントするマウントポイント55との3つの情報を含んでいる。マウント構成ファイル24には、I/Oパス名が“(ノード番号、I/Oインターフェイス番号、ディスクコントローラ番号)=(200, 290, 11)”のような形式で記述され、そのI/Oパスが使用可能な場合、マウント構成ファイル24の対応するI/Oパスの使用可否情報に“available”と記述され、そのI/Oパスが使用不可能な場合、使用可否情報に“unavailable”と記述される。図3に示した例では、I/Oパス(200, 290, 11)と(300, 390, 12)との両者がマウントポイント/mntに対応付けされており、共に使用可能となっている。この記述によって、ユーザが/mntディレクトリ以下のディレクトリツリー内のファイルにアクセスしたとき、物理ディスク装置10にアクセスできるようになる。このとき、物理ディスク装置10にアクセスするためのI/Oパスは、前述のいずれかのI/Oパスが使用される。使用していない方のI/Oパスは「待機中」状態としてスタンバイしている。

【0030】前述のように、物理ディスク装置にアクセスするためのI/Oパスが2つ存在する場合、その2つのI/Oパスを同じエントリに記載することにより、2つのI/Oパスを1つのマウントポイントに対応付けることができる。マウント構成ファイル24は、通常のエディタなどで編集することが可能であり、システム管理者は、システムの構成を変更したとき、マウント構成ファイル24の内容が新しいシステム構成と一致するように、マウント構成ファイル24を編集し、システムをリポートさせる。システムの起動時、ファイルサーバ140は、

修正後のマウント構成ファイル24に従ってマウント処理を行うので、リブート後、新しいシステム構成が使用可能となる。例えば、図1に示した物理ディスク装置装置20をシステムに追加したとき、システム管理者は“(200, 290, 21) available) (300, 390, 22) available) /mnt1”という行をマウント構成ファイル24に追加してシステムをリブートする。この記述によって、ユーザが /mnt1 ディレクトリにアクセスしたとき、前述の追加行に記載したいずれかのI/Oパスから物理ディスク装置20にアクセスできるようになる。

【0031】図4はシステムの起動時のファイルサーバの処理動作を説明するフローチャート、図5はシステム全体のノードの論理ディスク管理テーブルを更新する処理動作を説明するフローチャートであり、次に、これらのフローを参照して、システムの起動時にファイルサーバ140がマウント構成ファイル24を読み込み、論理ディスク管理テーブルを設定してマウント処理を行うまでの処理手順及び全ノードでの論理ディスク管理テーブルの更新の処理手順を説明する。

【0032】(1) システムの起動時、ノード100内のファイルサーバ140は、高信頼ディスク装置30上に格納されているマウント構成ファイル24の1つのエントリを読み込む(ステップ401、402)。

【0033】(2) ファイルサーバ140は、マウント構成ファイル24に記載されたI/Oパス名に対して論理ディスクIDを自動的に設定する。マウント構成ファイル24の1つのエントリに複数のI/Oパス名が記載されていた場合、ファイルサーバ140は、その複数のI/Oパスに対して1つの論理ディスクIDを設定する。例えば、図3に示した例の場合、ファイルサーバ140は、I/Oパス名5.1“(200, 290, 11)”及びI/Oパス名5.3“(300, 390, 12)”に対して論理ディスクID“123”を設定する。ファイルサーバ140は、これにより、設定した論理ディスクIDを論理ディスク管理テーブル170の論理ディスクID171に登録する(ステップ403)。

【0034】(3) 前述の第1のI/Oパス名をノード番号172、I/Oインターフェイス番号173、ディスクコントローラ番号174に登録し、第2のI/Oパス名をノード番号176、I/Oインターフェイス番号177、ディスクコントローラ番号178に登録する。図3に示した例の場合、論理ディスクID171には“123”、ノード番号172には“200”、I/Oインターフェイス番号173には“290”、ディスクコントローラ番号174には“11”、ノード番号176には“300”、I/Oインターフェイス番号177には“390”、ディスクコントローラ番号178には“12”が登録される(ステップ404)。

【0035】(4) そして、ファイルサーバ140は、

マウント構成ファイル24の使用可否情報に“available”と記載されている最初のI/Oパス“(200, 390, 11)”について、論理ディスク管理テーブル170の対応する状態フラグを「使用中」状態と登録し、“available”と記載されている残りのI/Oパス“(300, 390, 12)”について、対応する状態フラグを「待機中」状態と登録する。また、ファイルサーバ140は、マウント構成ファイル24の使用可否情報に、“unavailable”と記載されているI/Oパスについては対応する状態フラグを「使用不可」状態と登録する。この結果、論理ディスク管理テーブル170の内容は、図2に示したようなものとなる(ステップ405)。

【0036】(5) ファイルサーバ140は、マウント構成ファイル24に記載された全てのエントリについて、論理ディスク管理テーブル170への登録が終了したか否かをチェックし、終了していない場合、ステップ402からの処理を繰り返し実行して論理ディスク管理テーブルへの登録を続ける(ステップ406)。

【0037】(6) ステップ406で、マウント構成ファイル24に記載された全てのエントリについて、論理ディスク管理テーブル170への登録が終了していた場合、ファイルサーバ140は、全ての他のノード200、300であるリモートノードのファイルサーバと通信を行い、システムを構成する全ノードの論理ディスク管理テーブルの更新を行わせる(ステップ407)。

【0038】(7) ファイルサーバ140は、全リモートノードから論理ディスク管理テーブルの更新完了の通知を受信したら、マウント構成ファイル24に記載されているI/Oパス名“(200, 290, 11)”及び“(300, 390, 12)”とマウントポイント /mnt との対応関係、及び、論理ディスク管理テーブル170に登録した上記I/Oパス名と論理ディスクID“123”との対応関係から、マウントポイント /mnt と上記論理ディスクID“123”との対応関係を作り、論理ディスクID“123”に対応する論理ディスクをマウントポイント /mnt にマウントする(ステップ408)。

【0039】次に、図5に示すフローを参照して前述したステップ407の処理時のファイルサーバ140及びリモートノードのファイルサーバの処理動作を説明する。

【0040】(1) ファイルサーバ140は、自ノード100の論理ディスク管理テーブルの設定を終了した後、全リモートノードのファイルサーバに論理ディスク管理テーブル140の内容を送信し、論理ディスク管理テーブルを更新するように要求する(ステップ901、902)。

【0041】(2) この通知を受けたりリモートノードのファイルサーバは、送信されてきた論理ディスク管理テーブル170の内容を、そのノードの論理ディスク管理

テーブルに複写して論理ディスク管理テーブルの更新を行い、ファイルサーバ140に論理ディスク管理テーブルの更新終了を通知する(ステップ905~907)。

【0042】(3) ファイルサーバ140は、全リモートノードからそれぞれのノードの論理ディスク管理テーブルの更新完了通知を受信するのを待ち、図4により説明したステップ408のマウント処理を実行して処理を終了する(ステップ903、904)。

【0043】図6は通常運用時のファイルサーバの処理動作を説明するフローチャートであり、次に、このフローを参照して、通常運用時のファイルアクセスの手順について説明する。ここでは、ファイル管理テーブル160、260及び論理ディスク管理テーブル170、270の設定が図2に示すようになっており、ローカルノードとしてのノード200に接続された物理ディスク装置にアクセスする場合について、ノード200上で動作するユーザアプリケーション230が、ファイルID“100”を指定したファイルアクセス要求をファイルサーバ240に発行した場合を例に説明する。

【0044】(1) ファイルサーバ240は、ユーザアプリケーション230からの要求を受信すると、この要求が他のノードであるリモートノードからの要求であるかを判定する(ステップ501、502)。

【0045】(2) 説明している例では、自ノードであるローカルノードのユーザアプリケーションからのアクセスであるとしているので、ファイルサーバ240は、ファイル管理テーブル260を検索し、ファイルID“100”からそのファイルIDで定義されるファイルが格納されている論理ディスクの論理ディスクID“123”を求める(ステップ503)。

【0046】(3) そして、ファイルサーバ240は、論理ディスク管理テーブル270を検索し、論理ディスクIDから状態フラグが「使用中」状態のIOパス名“(200, 290, 11)”を求め、そのIOパス名に含まれるノード番号“200”がローカルノードであるかを判定する(ステップ504、505)。

【0047】(4) 前述のIOパス名に含まれるノード番号“200”がローカルノードであるとして説明しているため、ステップ505で、前述のIOパス名に含まれるノード番号“200”がローカルノードであると判定され、ファイルサーバ240は、自ローカルノードのディスクドライバ250にIOパスを指定したIOアクセス要求を送る。この要求を受けたディスクドライバ250は、IOインターフェイス290を介してディスクコントローラ11に制御信号を送る(ステップ507)。

【0048】次に、他のノードであるリモートノードに接続された物理ディスク装置にアクセスする場合について説明する。ここで説明する例は、ノード100上で動作するユーザアプリケーション130が、ファイルID

“100”を指定したファイルアクセス要求をファイルサーバ140に発行した場合であるとする。

【0049】(1) ファイルサーバ140は、ユーザアプリケーション130からの要求を受信すると、ローカルノードに接続された物理ディスク装置にアクセスする場合と同様に、ファイル管理テーブル160を検索しファイルID“100”から論理ディスクID“123”を求め、論理ディスク管理テーブル170を検索して論理ディスクID“123”からIOパス名“(200, 290, 11)”を求める(ステップ501~504)。

【0050】(2) ファイルサーバ140は、上記IOパス名に含まれるノード番号“200”がリモートノードであることを確認すると、そのノード(ノード200)のファイルサーバ240に上記論理ディスクIDを指定したIOアクセス要求を送る(ステップ505、506)。

【0051】(3) この要求を受けたファイルサーバ240は、論理ディスク管理テーブル270を検索し論理ディスクID“123”から状態フラグが「使用中」状態のIOパス名“(200, 290, 11)”を求める(ステップ501、502、504)。

【0052】(4) ファイルサーバ240は、IOパスに含まれるノード番号“200”が自ノードであるローカルノードであることを確認して、ディスクドライバ250にIOパスを指定したIOアクセス要求を送る。この要求を受けたディスクドライバ250は、IOインターフェイス290を介してディスクコントローラ11に制御信号を送る(ステップ505、507)。

【0053】前述した処理動作の説明から判るように、ファイルサーバが自ノードであるローカルノードからアクセス要求を受ける場合、その要求は、全てユーザアプリケーションからの要求であり、他のノードであるリモートノードからの要求を受ける場合、その要求は、全てリモートノードのファイルサーバからの要求である。

【0054】実際のファイルアクセス処理は、バッファキャッシュを経由して行われる。ファイルサーバ240は、論理ディスクIDを指定したIOアクセス要求に対する処理を、バッファキャッシュ280に対するリード・ライト処理と、バッファキャッシュ280と物理ディスク装置10との間でのリード・ライト処理とに分けて行う。ファイルサーバ240は、バッファキャッシュ280と物理ディスク装置10との間のリード・ライトアクセス処理との実行時に、論理ディスクIDからIOパス名への変換を行う。ノード100で動作するプログラムが、リモートノードに接続された物理ディスク装置10にアクセスする場合、ノード100上のバッファキャッシュ180とノード200上のバッファキャッシュ280とを経由してアクセスが行われる。すなわち、ライト処理を行う場合のデータの流れは、バッファキャッシ

ユ180→バッファキャッシュ280→物理ディスク装置10となる。リード処理の場合、この逆の順序となる。

【0055】ユーザアプリケーションがファイルを更新し、ファイルの更新日付が変わるなどして、ファイル管理テーブルの内容が変更されたとき、ファイル管理テーブルの変更を物理ディスク装置に書き戻す必要がある。次に、この書き戻し処理について説明する。

【0056】ファイル管理テーブルの内容が変更され、その内容をローカルノードに接続された物理ディスク装置に書き戻す場合、ローカルノードのファイルサーバがローカルノードのファイル管理テーブルの内容を直接その物理ディスク装置に書き戻す。また、リモートノードに接続された物理ディスク装置に書き戻す場合、ローカルノードのファイルサーバは、物理ディスク装置が接続されたノードにローカルノードのファイル管理テーブルの内容を一旦転送する。その後、物理ディスク装置が接続されたノードのファイルサーバが物理ディスク装置にその内容を書き戻す。例えば、ノード100のファイルサーバ140がファイル管理テーブル160の内容を物理ディスク装置10に書き戻す場合、まず、ファイルサーバ140は、物理ディスク装置への書き戻し処理を行いたいファイル管理テーブル160のエントリ中の、論理ディスクID162(“123”)を参照して、書き戻す先の論理ディスクIDを求める。そして、論理ディスク管理テーブル170を検索して上記論理ディスクIDに対応する物理ディスク装置にアクセスするためのIOPAS(“200, 290, 11”)を求め、そのIOPAS名に含まれるノード番号(“200”)に対応するノード(ノード200)のファイルサーバ240に書き戻しを行いたいファイル管理テーブルのエントリを送信する。ファイルサーバ240は、受信したデータを一旦ファイル管理テーブル260に書き込む。その後、ファイルサーバ240は、ファイル管理テーブルに保存されている他のデータと纏めて、ファイル管理テーブル260の更新内容を物理ディスク装置10に書き込む。ファイルサーバ240が物理ディスク装置10にアクセスするためのIOPASは、論理ディスク管理テーブル270を検索し、論理ディスクID262をIOPAS名に変換することによって求められる。

【0057】前述したように、最終的な物理ディスク装置へのデータの書き戻しは、物理ディスク装置が接続されたノードに存在するファイル管理テーブル及びバッファキャッシュから行っており、物理ディスク装置が接続されたノードのファイル管理テーブル及びバッファキャッシュには、ローカルノードのユーザアプリケーションに関係するもの以外にリモートノードのユーザアプリケーションに関係するものが存在する。

【0058】図7はIOPASの切り替えの処理動作を説明するフローチャート、図8～図10はIOPASに障害

が発生しIOPASの切り替えを行う処理について説明する図である。図8～図10において、13はディスクキャッシュ、340はファイルサーバ、350はディスクドライバ、360はバッファキャッシュであり、他の符号は図1の場合と同一である。以下、これらの図を参照して、ディスクコントローラ11で障害が発生し、通常使用しているIOPAS(“200, 290, 11”)が使用不可能になったとき、物理ディスク装置10にアクセスするためのIOPASを(“200, 290, 11”)から(“300, 390, 12”)に切り替える処理について説明する。

【0059】図9において、ディスクキャッシュ13は、ディスク装置10が備えるディスクコントローラ11の内部に設けられたディスクキャッシュであり、ディスクコントローラ11に対してリード・ライト処理要求が発行されたときに使用される。そして、実際のリード・ライト処理は、このディスクキャッシュ13を経由して行われる。また、ディスクコントローラ12は、ディスクコントローラ11に障害が発生したときに、ディスクキャッシュ13がディスク媒体に書き戻す必要のあるデータを保持している場合、そのデータをディスク媒体に書き戻し、ディスクコントローラ11をディスク装置から切り放す機能を持つ。

【0060】図8は図7により説明するステップ1003でのリクエストの保留の処理を行うときの各ノードの動作を示し、図9は図7により説明するステップ1004でのディスクキャッシュの書き戻しの処理と、ステップ1005でのバッファキャッシュの転送の処理を行うときの各ノードの動作を示し、図10は図7により説明するステップ1006でのリクエストの保留解除及び転送の処理を行うときの各ノードの動作を示している。

【0061】以下、ディスクコントローラ11で障害が発生した時、物理ディスク装置10にアクセスするためのIOPASを(“200, 290, 11”)から(“300, 390, 12”)に切り替える処理を図8～図10を併用しながら図7に示すフローを参照して説明する。なお、論理ディスク管理テーブル270の設定は図2に示すようになっているものとする。

【0062】障害検出の処理(ステップ1001) ディスクコントローラ11に障害が発生すると、ディスクドライバ250は、IOPAS(200, 290, 11)を使って物理ディスク装置10にアクセスを行うことができなくなる。これをもって障害検出とし、ディスクドライバ250は、IOPAS(200, 290, 11)の障害発生をファイルサーバ240に通知する。また、ディスクドライバ250がローカルノードとしてのノード200のノード番号を含むIOPASのうち、論理ディスク管理テーブル270の状態フラグが、「使用中」状態及び「待機中」状態のIOPASを定期的に監視することによって障害を検出してよい。これによ

て、「待機中」状態のI/Oパスの障害検出が可能となる。

【0063】切り替え対象I/Oパスの検索の処理（ステップ1002）

障害発生通知を受けたファイルサーバ240は、図2に示した論理ディスク管理テーブル270を参照し、障害発生I/Oパス“(200, 290, 11)”を含むエントリを検索する。そして、障害発生I/Oパスの状態フラグが「待機中」状態であるか否かをチェックし（ステップ1010）、もし、障害発生I/Oパスの状態フラグが「待機中」状態であれば、I/Oパスの切り替え処理は必要なく、ステップ1011の処理に進む。そうでない場合、I/Oパスの切り替えが必要になりステップ1103の処理に進む。前述の検索によって見つかったエントリには、障害発生I/Oパス以外に、状態フラグ279（275）が「待機中」状態のI/Oパス“(300, 390, 12)”と論理ディスクID“123”が登録されている。この「待機中」状態のI/Oパス“(300, 390, 12)”が切り替え先のI/Oパスとなる。ファイルサーバ240は、障害発生I/Oパス名と切り替え先のI/Oパス名とそれらに対応する論理ディスクID（以後、I/Oパス切り替え処理を行う論理ディスクIDと呼ぶ）を、ファイルサーバ240が管理するメモリ内に保存し、ファイルサーバ240が論理ディスク管理テーブル270を検索することなくいつでも得られるようにしておく。

【0064】リクエストの保留の処理（ステップ1003）

この処理について、図8を参照して説明する。ファイルサーバ240は、現在処理中あるいは今後受理するI/Oアクセス要求の中で、I/Oパスの切り替え処理を行う論理ディスクID“123”あるいは障害発生I/Oパス“(200, 290, 11)”を指定したI/Oアクセス要求を保留し、その内容を後で取り出すことができるようにファイルサーバ240が管理するメモリ上に記録する。図8に示す例では、ファイルサーバ140は、ディスクコントローラ11で障害が発生したことを知らずに、論理ディスク“123”を指定したライト要求をファイルサーバ240に送信している。ファイルサーバ240は、このライト要求と、現在処理中のI/Oパス名“(200, 290, 11)”を指定したリード要求を保留している。

【0065】次に、ファイルサーバ240は、切り替え先のI/Oパス“(300, 390, 12)”に含まれるノード番号“300”に対応するノード（以後、切り替え先のノードと呼ぶ）のファイルサーバ340に、障害発生I/Oパス名“(200, 290, 11)”と切り替え先のI/Oパス名“(300, 390, 12)”と対応する論理ディスクID“123”とを送信し、論理ディスクIDを指定したI/Oアクセス要求を保留するように

要求する。この要求を受信したファイルサーバ340は、前述の2つのI/Oパス名と論理ディスクIDとをファイルサーバ340が管理するメモリ上に保存し、これらの情報をいつでも得られるようにした後、論理ディスクID“123”を指定したI/Oアクセス要求を保留し、その内容を後で取り出せるようにファイルサーバ340が管理するメモリ上に保存する。図8に示す例では、ファイルサーバ340は、論理ディスクID“123”を指定したリード要求を保留している。

【0066】ディスクキャッシュの書き戻しの処理（ステップ1004）

この処理について、図9を参照して説明する。ファイルサーバ340は、リクエストの保留の設定を行った後、障害発生I/Oパスを含むディスクコントローラ番号“11”に対応するディスクコントローラ11が備えるディスクキャッシュ13を、切り替え先のI/Oパスを含むディスクコントローラ番号“12”に対応するディスクコントローラ12を使ってディスク装置に書き戻すようにディスクドライバ350に要求する。この要求を受けたディスクドライバ350は、I/Oインターフェイス390を介してディスクコントローラ12に制御信号を送りディスクキャッシュ13に保存されているdirtyなデータをディスク領域に書き戻し、ディスクコントローラ11をディスク装置10から切り放す。これらの処理の終了後、ディスクドライバ350は、ファイルサーバ340に終了通知を送る。

【0067】バッファキャッシュの転送の処理（ステップ1005）

この処理について、図9を用いて説明する。ファイルサーバ340は、ディスクドライバ350からの終了通知を受けると、障害発生I/Oパス“(200, 290, 11)”に含まれるノード番号“200”に対応するノード（以後、障害発生ノードと呼ぶ）のファイルサーバ240にファイル管理テーブル260及びバッファキャッシュ280の転送を要求する。ファイルサーバ340からの要求を受信したファイルサーバ240は、dirtyな（物理ディスク装置に書き戻す必要のある）ファイル管理テーブル260とdirtyなバッファキャッシュ280の中で、論理ディスクID262や論理ディスクID281が、I/Oパス切り替え処理を行う論理ディスクID“123”であるデータを、ファイルサーバ340に送信する。この送信が成功したら、ファイルサーバ240は、ノード200内に存在する前述のデータを消去可能とし、バッファキャッシュ280をしばらくの間、読み出し用のキャッシュとして使用するが、バッファキャッシュ280やファイル管理テーブル260のためのメモリ領域が不足してきたらこれらを消去する。ファイルサーバ340は、受け取ったデータを、ノード300上のファイル管理テーブル360及びバッファキャッシュ380にマージする。ノード300上のこれらの

データはdirtyであるので、I/Oパスの切り替え処理が終了し通常運用状態となったら、ファイルサーバ340が切り替え先のI/Oパス“(300, 390, 12)”を使用して物理ディスク装置10に書き込む。また、前述の上記データは、読み出し用のキャッシュとして使用される可能性もある。

#### 【0068】論理ディスク管理テーブルの更新の処理 (ステップ1006)

この処理は、図5により説明したフローの手順で実行される。図5に示したローカルノードは、ここでは障害発生ノード200である。ファイル管理テーブル260及びバッファキャッシュ280の転送が終了すると、ファイルサーバ240は、論理ディスク管理テーブル270に登録されている障害発生I/Oパス“(200, 290, 11)”の状態フラグ275を「使用中」状態から「使用不可」状態に、切り替え先のI/Oパス“(300, 390, 12)”の状態フラグ279を「待機中」状態から「使用中」状態に更新する。ファイルサーバ240は、論理ディスク管理テーブル270の更新の終了後(図5のステップ901)、全リモートノードのファイルサーバに論理ディスク管理テーブル270の更新情報を送り、論理ディスク管理テーブルの更新を要求し

(図5のステップ902)、リプライを待つ。例えば、ファイルサーバ240からの要求を受信したノード100のファイルサーバ140は、受信した論理ディスク管理テーブル270の更新情報に基づいて、ノード100の論理ディスク管理テーブル170のI/Oパス“(200, 290, 11)”に対応する状態フラグ175を「使用不可」状態に、I/Oパス“(300, 390, 12)”に対応する状態フラグ179を「使用中」状態に更新する(図5のステップ906)。この更新の後、ファイルサーバ140は、ファイルサーバ240に論理ディスク管理テーブル370の更新終了の通知を送る(図5のステップ907)。ファイルサーバ240が、全リモートノードのファイルサーバから論理ディスク管理テーブルの更新終了の通知を受信すれば(図5のステップ903)、システムを構成するすべてのノードの論理ディスク管理テーブルの更新が完了したことになる。

#### 【0069】リクエストの保留解除及び転送の処理 (ステップ1007)

この処理について、図10を参照して説明する。ファイルサーバ240は、切り替え先のノードのファイルサーバ340にリクエストの保留を解除する要求を送る。この要求を受けたファイルサーバ340は、ステップ1003で行ったI/Oアクセス要求の保留を解除し、保留していたI/Oアクセス要求の処理を行い、通常運用時の処理を開始する。また、ファイルサーバ240は、ステップ1003で行ったI/Oアクセス要求の保留を解除し、保留していたI/Oアクセス要求のうち、障害発生I/Oパスを指定したI/Oアクセス要求を、切り替え先のI/Oパ

スを指定したI/Oアクセス要求に変換した後、保留中のすべてのI/Oアクセス要求を、切り替え先のノードのファイルサーバ340に転送する。図10に示す例では、ファイルサーバ240は、I/Oパス“(200, 290, 11)”を指定したリード要求を、I/Oパス“(300, 390, 12)”を指定したリード要求に変換し、前述の要求と論理ディスクID“123”を指定したライト要求とをノード300のファイルサーバ340に転送している。転送されたI/Oアクセス要求は、ファイルサーバ340によって処理される。

#### 【0070】マウント構成ファイルの更新の処理 (ステップ1008)

最後に、ファイルサーバ240は、高信頼ディスク装置30が接続されているノード100のファイルサーバ140に障害発生I/Oパス“(200, 290, 11)”が「使用不可」状態になったことをマウント構成ファイル24に記載するように要求し、通常運用時の処理を開始する。この要求を受けたファイルサーバ140は、高信頼ディスク装置30上のマウント構成ファイル24を参照し、障害発生I/Oパス“(200, 290, 11)”の使用可否情報52を“unavailable”(使用不可)に書き換える。以上により、I/Oパスの切り替え処理が終了する。

#### 【0071】論理ディスク管理テーブルの更新の処理 (ステップ1011)

ステップ1010のチェックで、I/Oパスの切り替え処理を行う必要がなかった場合、障害発生ノードのファイルサーバ240は、ステップ1006の処理と同様の手順でシステム全体の論理ディスク管理テーブルを更新する。但し、障害発生I/Oパス“(200, 290, 11)”の状態フラグを「待機中」から「使用不可」に書き換える処理だけを行う。システム全体の論理ディスク管理テーブルの更新が終了した後前述したステップ1008の処理に進む。

【0072】図11はI/Oパスが障害から復旧したとき、I/Oパスをシステムに復旧させる処理手順を説明するフローチャートであり、これについて説明する。ここでは、物理ディスク装置10のディスクコントローラ11の障害などの原因により「使用不可」状態になっていたI/Oパス“(200, 290, 11)”がディスクコントローラ11の交換などによって再び使用可能になったとき、システムに上記のI/Oパスを復旧させる方法を例に説明する。また、ここでは、I/Oパスの復旧処理中に使用中のI/Oパスに障害が発生することはないと仮定する。

【0073】(1) 障害が発生したディスクコントローラの交換等により、今まで使用不可能となっていたI/Oパス“(200, 290, 11)”が使用可能な状態になると、システム管理者は、管理用のプログラムを使って、このI/Oパスをシステムに復旧させる要求を、高信

頼ディスク装置が接続されているノード100のファイルサーバ140に送信する。ファイルサーバ140は、この要求を受信する(ステップ601)。

【0074】(2) 復旧要求を自したファイルサーバ140は、論理ディスク管理テーブル170を参照して、前述のIOパス“(200, 290, 11)”の状態フラグ175を「使用不可」状態から「待機中」状態に更新する。また、ファイルサーバ140は、論理ディスク管理テーブル170の更新が終了したら、全ての稼働中のノードのファイルサーバと通信を行い、全ノードの論理ディスク管理テーブルを論理ディスク管理テーブル170と同じ内容にする。この処理は、図7によるIOパスの切り替えのフローにより説明したステップ1006での処理と同様な処理により行われる(ステップ602)。

【0075】(3) そして、ファイルサーバ140は、高信頼ディスク装置30上のマウント構成ファイル24を参照し、前述のIOパス“(200, 290, 11)”の使用可否情報52を“unavailable”(使用不可)から“available”(使用可)に変更する。前述の処理により、IOパス“(200, 290, 11)”を「待機中」状態としてシステムに復旧させることができる(ステップ603)。

【0076】前述した本発明の実施形態は、ファイル管理テーブル260及びバッファキャッシュ270をノード200からノード300に転送するとして説明した(図7のステップ1005)が、これは次のような理由による。すなわち、物理ディスク装置へのアクセスは、ローカルノードからのアクセスでもリモートノードからのアクセスでも、最終的に、その物理ディスク装置が接続されたノードのファイル管理テーブル及びバッファキャッシュを経由して行われる。従って、物理ディスク装置が接続されたノードは、そのノード(ローカルノード)で動作するプログラムに関係するファイル管理テーブル及びバッファキャッシュの他に、リモートノードで動作するプログラムに関係するファイル管理テーブル及びバッファキャッシュを持つ。前述した本発明の実施形態に示したようなIOパス切り替え処理は、物理ディスク装置が接続されているノードがノード200からノード300に切り替わるので、ノード300がノード200に代わって、ノード200が保持していたファイル管理テーブル260及びバッファキャッシュ280を持つ必要がある。そこで、IOパス切り替え処理時にファイル管理テーブルやバッファキャッシュをノード300に転送するようにしている。このとき、dirtyなデータのみを転送するようにして、データの転送量をなるべく少なく済むようにしている。

【0077】また、前述した本発明の実施形態は、物理ディスク装置10、20を共にノード200から使用しているときに、IOインターフェイス290に障害が発

生した場合、IOパス(200, 290, 11)及び(200, 290, 21)の両方が使用できなくなるが、この場合、ディスクドライバ250が、各々のIOパスに対して障害検出を行い、各々のIOパスに対して前述の各ステップで示されるIOパスの切り替え処理を行うようにすればよい。また、ディスクドライバ250がIOインターフェイス290で障害が起こったことを検出する機能を持つ場合、ステップ1001で、ディスクドライバ250がファイルサーバ240にIOインターフェイス290の障害を通知し、ステップ1002で、ファイルサーバ240が論理ディスク管理テーブル270を検索し、障害発生IOインターフェイス番号“290”から、障害発生IOパス(200, 290, 11)、(200, 290, 21)と対応する切り替え先のIOパスと論理ディスクIDを探し出し、これら2組のIOパスについて、前述の各ステップで示される切り替え処理を同時に行うようにしてもよい。

【0078】前述した本発明の実施形態において、ノード200が2つのIOインターフェイスを有し、物理ディスク装置10がこれら2つのIOインターフェイスによってノード200と接続されており、物理ディスク装置10とノード200との間のIOパスが2つ存在し、通常運用時これらのIOパスのうち1つを利用しているような場合、ディスクコントローラやIOインターフェイスの障害発生により、今まで使用していたIOパスが使用できなくなったとき、物理ディスク装置10にアクセスするためのIOパスをもう片方のIOパスに前述したの方法で切り替えることができる。この場合、ステップ1003でノード300のファイルサーバ340がIOアクセス要求を保留する処理と、ステップ1005でノード200が持つバッファキャッシュ280及びファイル管理テーブル260をIOパス切り替え先のノード300に転送する処理が不要となる。

【0079】また、本発明は、物理ディスク装置にアクセスするためのIOパスが3つ以上存在する場合にも適用することができる。この場合、論理ディスク管理テーブル及びマウント構成ファイル24の各エントリに3つ以上のIOパスの組を登録できるようにし、システムの起動時にファイルサーバ140がマウント構成ファイル24に記載されたIOパスの組に対して、1つの論理ディスクIDを設定し、IOパスと論理ディスクIDとの対応関係を論理ディスク管理テーブルに登録するようにすればよい。そして、この場合、通常運用時、複数のIOパスが「待機中」状態としてスタンバイするため、障害発生時のIOパスの切り替え処理を行う際に、複数の「待機中」状態のIOパスの中から切り替え先のIOパスを選択する必要がある。この切り替え先のIOパスの決定は、前述した実施形態におけるステップ1002で障害を検出したノードのファイルサーバがそのノードの論理ディスク管理テーブルを検索し、障害発生IOパス

名を含むエントリを見つけたときに、そのエントリのなるべく最初の方のフィールドに登録されている「待機中」状態のI/Oバスを切り替え先のI/Oバスとして選出することによって行うようにすればよい。また、論理ディスク管理テーブルに登録されている各I/Oバス毎に使用時間（状態フラグが「使用中」状態となっていた時間）を上記論理ディスク管理テーブルに登録できるようにし、I/Oバスの切り替え処理時、使用時間の短いI/Oバスに切り替えるようにしてもよい。これによって、複数のI/Oバスをまんべんなく使用することができる。

【0080】さらに、本発明は、LAN等のネットワークにより接続された疎結合計算機システムによるファイルシステムに対しても適用することができる。この場合、前述のノード番号の代わりにネットワークアドレスを使用すればよい。

【0081】また、前述した本発明の実施形態において、ディスクキャッシュ13をディスクコントローラ12から制御し、ディスク装置10に書き戻す機能を物理ディスク装置10が持たない場合、ノード200のディスクドライバ250が、ディスクキャッシュ13に保存されたdirtyなキャッシュを少なくとも含むデータを予め保持しておいて、障害発生時、前述のステップ1004でディスクドライバ250がディスクドライバ350と通信を行い、dirtyなディスクキャッシュを少なくとも含むようなデータをノード200からノード300に転送し、ディスクコントローラ12を通してディスク装置10に書き戻すようにしてもよい。

【0082】前述した本発明の実施形態は、I/Oバス切り替え処理中、障害発生ノード及び切り替え先のノードに送信されてきたI/Oアクセス要求は、保留するようにしていたが、I/Oアクセス要求を保留しないようにすることもできる。以下、この場合のファイルサーバの動作について図面により説明する。

【0083】図12はI/Oバス切り替え時の障害発生ノードの処理動作の他の例について説明するフローチャート、図13は障害発生ノード以外のノードの処理動作の他の例を説明するフローチャートである。以下、障害発生ノードがノード200、切り替え先のノードがノード300の場合を例として、図12、図13に示すフローを参照して、I/Oバス切り替え処理中に各ノードに送信されてきたI/Oアクセス要求の処理の方法を説明する。まず、障害発生ノードのファイルサーバの動作を図12のフローにより説明する。

【0084】（1）障害発生ノードのファイルサーバ240は、I/Oバス切り替え処理中に、I/Oアクセス要求を受信すると、その要求が他のノードであるリモートノードからの要求が否かを判定する（ステップ701、702）。

【0085】（2）ステップ702の判定で、受信したI/Oアクセス要求がローカルノード（自ノード）のユー

ザアプリケーション230からのものであると判定すると、ファイルサーバ240は、前述した実施形態で説明したと同様に、I/Oバス切り替え処理の間、その要求を保留する。この要求は、I/Oバスの切り替え処理終了時に、切り替え先のノードに送信される（ステップ703）。

【0086】（3）ステップ702の判定で、受信したI/Oアクセス要求がリモートノードからのものであると判定すると、ファイルサーバ240は、その要求に対してリプライを返さずに無視する（ステップ704）。

【0087】次に、障害発生ノード以外のノードのファイルサーバの動作を図13に示すフローを参照して説明する。障害発生ノード以外のファイルサーバは、基本的に図4により説明した通常運用時と同様の動作をするので、ここでは図4の処理と重なる部分については説明を省略する。

【0088】（1）障害発生ノード以外のファイルサーバがI/Oバス切り替え中に障害発生ノード（ノード200）に送信したI/Oアクセス要求はタイムアウトとなる（ステップ808）。

【0089】（2）I/Oアクセス要求がタイムアウトになったら、I/Oアクセス要求を送信したファイルサーバは、一定時間（例えば1秒）待った後、論理ディスク管理テーブルを参照して、論理ディスクIDからI/Oバス名を求める処理から処理をやり直す。このとき、I/Oバスの切り替え処理が終了していれば、全ノードの論理ディスク管理テーブルが更新されているので、ステップ804の処理によって切り替え先のI/Oバスが求まる（ステップ804）。

【0090】（3）I/Oアクセス要求を送信しようとしているファイルサーバは、求められたI/Oバス名が含むノードがローカルノードであるか否かを判定し、切り替え先のI/Oバス名が含むノードがローカルノードでなかった場合、I/Oアクセス要求を切り替え先のノード（ノード300）に送信する（ステップ805、806）。

【0091】（4）ステップ805の判定で、切り替え先のI/Oバスがローカルノードであれば、I/Oアクセス要求を送信しようとしているファイルサーバは、I/Oアクセス要求をローカルノードのディスクドライバに送信する（ステップ807）。

【0092】前述したステップ804の処理において、もし、論理ディスクIDからI/Oバス名を求めなおしたときに、I/Oバスの切り替え処理が終了していない場合、I/Oアクセス要求は、障害発生ノード（ノード200）に送信され、上記I/Oアクセス要求は再びタイムアウトとなり、I/Oアクセス要求が成功するまで前述した処理が繰り返される。

【0093】この方法を使用することにより、図7により説明したステップ1003のリクエストの保留処理でリモートノードからのアクセス要求を保留する必要がな



くなるので、I/Oアクセス要求を保留するためのメモリを節約することができる。また、I/Oアクセス要求の再送回数に制限(例えば5回)を設け、もし制限回数だけ再送を行ってもタイムアウトになり続けられれば、そのI/Oアクセス要求をエラーとしてもよい。また、I/Oパス切り替え処理中、障害発生ノードのファイルサーバ240は、リモートノードからのI/Oアクセス要求を無視するかわりに、「I/Oパス切り替え処理中なので、I/Oアクセス要求を処理できない」という意味の通知をアクセス要求を送信したリモートノードのファイルサーバに送信するようにしてもよい。これにより、リモートノードのファイルサーバは、I/Oパスで障害が発生した場合とノード200で障害が発生した場合とを区別することができるようになる。

【0094】前述までに説明した本発明の第1の実施形態によるI/Oパス切り替え方法は、ノード200でOSの障害が発生したとき、ネットワーク1を通じてバッファキャッシュ280やファイル管理テーブル260をノード300に転送することができなくなるため、同じ方法でI/Oパスの切り替えを行うことは不可能である。

【0095】これを解決するため、本発明は、バッファキャッシュ280やファイル管理テーブル260をノード300に転送するための専用のハードウェアを使う方法を取ることができる。以下、これを第2の実施形態として説明する。

【0096】図14は本発明の第2の実施形態によるディスクキャッシュの書き戻しの処理とバッファキャッシュの転送の処理とを説明する図である。

【0097】本発明の実施形態におけるI/Oパス切り替え処理の手順は、前述までに説明した第1の実施形態の場合の図7に示すフローと同様に行われる。但し、第2の実施形態では、ステップ1003及びステップ1007の処理は行わない。そして、図14には、ステップ1004でのディスクキャッシュの書き戻しの処理とステップ1005でのバッファキャッシュの転送の処理についてしめしている。

【0098】図14において、メモリアクセス手段299(399)は、ノード200(300)に付属しており、メモリアクセス手段299とメモリアクセス手段399とは専用通信線2によって互いに接続されている。メモリアクセス手段299は、ノード200でOSの障害が発生しノード200上で動作するプログラムの全てが停止した場合にも、メモリ220にアクセスし、その内容を専用通信線2を使用してメモリアクセス手段399との通信によりノード300に送信することが可能なハードウェアである。

【0099】通常運用時、図14に示す各ノードのファイルサーバは、図13により説明した動作を行う。ここで例えば、ノード200でOSの障害が発生したとすると、あるファイルサーバがノード200に送信したI/O

アクセス要求のリプライが戻ってこないで、I/Oアクセスを送信したファイルサーバは、上記I/Oアクセス要求をタイムアウトにする(ステップ808)。ファイルサーバは、一定時間待った後、ローカルノードの論理ディスク管理テーブルを参照し、論理ディスクIDからI/Oパスを求める処理から処理の再実行を行うことになる(ステップ804)。I/Oパス切り替え処理中、前述の要求は、障害発生ノード(ノード200)に送信されタイムアウトとなるが、I/Oパス切り替え終了後、要求は切り替え先のノードに送信される。

【0100】以下、ノード200で障害が発生しノード200で動作する全てのプログラムが停止した場合に、物理ディスク装置10にアクセスするためのI/Oパスを(200, 290, 11)から(300, 390, 12)に切り替えるものとして、その処理を図1、図2、図14を併用しながら図7に示すフローを参照して説明する。

【0101】障害検出の処理(ステップ1001)ノード200で障害が発生すると、ノード200は、リクエストを一切受け付けなくなる。従って、ノード200にI/Oアクセス要求を送信したリモートノードのファイルサーバは、I/Oアクセス要求をタイムアウトとする。I/Oアクセス要求を送信したファイルサーバは、このタイムアウトによってノード200で障害が発生したことを検出する。前述したように、I/Oアクセス要求を送信したファイルサーバは、I/O処理要求がタイムアウトになったらその要求を再送するので、何度も障害発生ノード(ノード200)に上記要求を再送し、そのたびに要求をタイムアウトにする可能性がある。上記ファイルサーバは、あるノードへの要求が最初にタイムアウトになったとき、次のステップ1002の処理に進み、2回目以降、ステップ1002以降の処理は行わない。

【0102】切り替え対象I/Oパスの検索の処理(ステップ1002)

I/Oアクセス要求を送信したファイルサーバは、ローカルノードの論理ディスク管理テーブルを参照し、障害が発生したノードのノード番号“200”から障害発生I/Oパス名と切り替え先のI/Oパス名とを探し出し、切り替え先のI/Oパスが含むノード番号に対応するノード

(切り替え先のノード)のファイルサーバに、障害発生I/Oパスから切り替え先のI/OパスにI/Oパスを切り替えるように要求する。切り替え先のノードがローカルノード(自ノード)であれば、I/Oアクセスを送信したファイルサーバは、直ちにI/Oパスの切り替えの処理を開始する。但し、障害発生I/Oパスの状態フラグが「待機中」状態の場合(ステップ1010)、I/Oパスの切り替え処理は必要なくステップ1011の処理に進む。例えば、ノード100のファイルサーバ140がノード200のファイルサーバ240に送信したI/O処理要求がタイムアウトとなった場合、ファイルサーバ140は、

図2に示した論理ディスク管理テーブル170を検索し、ノード番号“200”を含むエントリを探す。見つかったエントリには複数のI/Oパスが記載されているが、ノード番号“200”を含むI/Oパス“(200, 290, 11)”が障害発生I/Oパスであり、状態フラグが「待機中」状態でノード番号“200”を含まないI/Oパス“(300, 390, 12)”が切り替え先のI/Oパスである。障害発生I/Oパスの状態フラグ275が「使用中」状態であるので、ファイルサーバ140は、切り替え先のノード300のファイルサーバ340に“(200, 290, 11)”から“(300, 390, 12)”にI/Oパスを切り替えるように要求する。もし、上記障害発生I/Oパスの状態フラグが「待機中」状態であれば、I/Oパスの切り替え処理は必要なく、ステップ1011の処理に進む。

【0103】前述した検索処理で、切り替え処理を行うI/Oパスの組が複数個見つかった場合、障害を検出したファイルサーバは、I/Oパス毎に対応する切り替え先のノードのファイルサーバにI/Oパスの切り替え要求を送信する。但し、複数のI/Oパスの切り替え要求を1つのノードに送る必要がある場合、それらのI/Oパスの切り替え要求を一括して送り、切り替え先のノードのファイルサーバが、それらのI/Oパスの切り替え処理を同時に行う。例えば、物理ディスク装置10と物理ディスク装置20とをノード200から使用していた場合、ノード200の障害を検出したファイルサーバは、ノード300のファイルサーバ340に上記2つの物理ディスク装置にアクセスするための2組のI/Oパスを切り替える要求を発行し、ファイルサーバ340は、前述した2組のI/Oパスの切り替え処理を同時に行う(ステップ1004~1008)。

【0104】ディスクキャッシュの書き戻しの処理(ステップ1004)

障害発生I/Oパス“(200, 290, 11)”から切り替え先のI/Oパス“(300, 390, 12)”にI/Oパスを切り替えるように要求されたファイルサーバ340は、I/Oパスの切り替えモードに入り、その後再び同じI/Oパス切り替え要求が送られてきても受理しない。これによって、I/Oパスの切り替え処理が二重に行われることを防止する。このステップの処理の後の処理内容は、第1の実施形態の場合と同様に行われる。ファイルサーバ340は、図14に示すように、ディスクドライバ350にディスクキャッシュの書き戻し要求を送信することにより、ディスクキャッシュ13の内容をディスク領域に書き戻して、ディスクコントローラ11を物理ディスク装置から切り放す。

【0105】バッファキャッシュの移動の処理(ステップ1005)

ファイルサーバ340は、次に、図14に示すように、メモリアクセス手段399に、障害が発生したノード2

00のファイル管理テーブル260とバッファキャッシュ280との内容をローカルノード(ノード300)に転送するように要求する。メモリアクセス手段399は、メモリアクセス手段299と通信を行い、専用通信線2を介して、dirtyなバッファキャッシュ280及びdirtyなファイル管理テーブル260の内容をノード300のファイルサーバ340に転送する。ファイルサーバ340は、ノード300上のファイル管理テーブル360及びバッファキャッシュ380にメモリアクセス手段399から送られてきたデータをマージする。マージされたデータは、I/Oパスの切り替え終了後、ファイルサーバ340によって切り替え先のI/Oパスから物理ディスク装置10に書き込まれる。また、これらデータは、読み出し用のキャッシュとしても使われる可能性もある。

【0106】論理ディスク管理テーブルの更新の処理(ステップ1006)

データの転送処理が終了した後、ファイルサーバ340は、論理ディスク管理テーブル370に登録されているI/Oパスの状態フラグを、障害発生I/Oパス“(200, 290, 11)”について、「使用不可」状態に、切り替え先のI/Oパス“(300, 390, 12)”について、「使用中」状態に登録し直す。ファイルサーバ340は、論理ディスク管理テーブル370の更新の終了後、第1の実施形態の場合と同様な方法により、全ての稼働中のノードのファイルサーバと通信を行うことにより、全ての稼働中のノードの論理ディスク管理テーブルに登録されている、障害発生I/Oパスの状態フラグを「使用不可」状態に、切り替え先のI/Oパスの状態フラグを「使用中」状態に更新する。

【0107】マウント構成ファイルの更新の処理(ステップ1008)

ファイルサーバ340は、全ての稼働中のノードの論理ディスク管理テーブルの更新が終了した後、高信頼ディスク装置30が接続されているノード100のファイルサーバ140に、I/Oパス“(200, 290, 11)”が「使用不可」状態になったことをマウント構成ファイル24に記載するように要求し、I/Oパスの切り替えモードから抜け、通常運用時の処理を開始する。前述の要求を受けたファイルサーバ140は、「使用不可」状態となったI/Oパス“(200, 290, 11)”の使用可否情報52を“available”(使用可)から“unavailable”(使用不可)に更新する。以上によりI/Oパスの切り替え処理が終了する。

【0108】論理ディスク管理テーブルの更新の処理(ステップ1011)

ステップ1010で、障害発生パスが「待機中」状態にあると判定され、I/Oパスの切り替え処理を行う必要がない場合、ステップ1001の処理で障害を検出したファイルサーバは、ステップ1006の処理と同様の手順

でシステム全体の論理ディスク管理テーブルを更新する。但し、障害発生 I/Oパスの状態フラグを「使用不可」に書き換える処理だけを行う。システム全体の論理ディスク管理テーブルの更新が終了した後、前述のファイルサーバがファイルサーバ 140 に対してマウント構成ファイルの更新を要求し、この要求を受けたファイルサーバ 140 は、ステップ 1008 の処理を行う。

【0109】図 15 は本発明の第 3 の実施形態によるファイルシステムの構成を示すブロック図、図 16 は本発明の第 3 の実施形態におけるマウント構成ファイルの具体的な構成例を説明する図であり、図 15 における符号は図 1 の場合と同一である。図 15 に示す本発明の第 3 の実施形態は、同一のファイルを物理ディスク装置 10 と物理ディスク装置 20 とに二重化（ミラーリング）して記録する例である。

【0110】図示本発明第 3 の実施形態において、マウント構成ファイルの 1 つのエントリには、図 16 に示すように、物理ディスクにアクセスするための I/Oパス名 51、53、各 I/Oパスの使用可否情報 52、54、マウントポイント 55 が記載されている。この第 3 の実施形態は、マウントポイントの 1 つのエントリに記載された I/Oパスからアクセスされる物理ディスク装置にファイルが多重化して記録される。従って、前述の I/Oパスからアクセスされる物理ディスク装置は異なるものがある必要がある。図 16 に示す例では、/mnt ディレクトリ以下のディレクトリに格納されたファイルは、I/Oパス“(200, 290, 11)”、“(300, 390, 22)”からアクセスされる物理ディスク装置（物理ディスク装置 10、20）にミラーリングされる。このような指定方法を採用することにより、システム管理者が論理ボリュームの設定を行う必要がなくなる。

【0111】システム立ち上げ時、ファイルサーバ 140 は、マウント構成ファイル 24 を読み込んで、第 1 の実施形態の場合と同様の手順で、全てのノードの論理ディスク管理テーブルを設定する。但し、第 3 の実施形態では、ファイルサーバ 140 は、マウント構成ファイル 24 の使用可否情報に“available”（使用可）と記載されているすべての I/Oパスについて、論理ディスク管理テーブルの対応する状態フラグに「使用中」と登録する。

【0112】次に、通常運用時のファイルサーバの動作を、ノード 100 のユーザアプリケーション 130 がファイル ID “100” を指定したファイルアクセス要求をファイルサーバ 140 に発行した場合を例に、図 15、図 16 を参照し、図 6 に示すフローに基づいて説明する。なお、ファイル管理テーブルの設定は図 2、論理ディスク管理テーブルの設定は図 16 に示すようになっているものとする。

【0113】(1) ファイルサーバ 140 は、ユーザアプリケーション 130 がファイル ID を指定したアク

セス要求を受けると、その要求がリモートノードからの要求であるか否かを判定し、自ノードからの要求である場合、ファイル管理テーブル 160 を検索し、ファイル ID “100” から論理ディスク ID “123” を求める（ステップ 501～503）。

【0114】(2) そして、ファイルサーバ 140 は、論理ディスク管理テーブル 170 を検索し、論理ディスク ID “123” から状態フラグが「使用中」状態の I/Oパス名“(200, 290, 11)”、“(300, 390, 22)”を求める（ステップ 504）。

【0115】(3) アクセス要求がライト要求の場合は、前述の両方の I/Oパスに対して同一内容の書き込みを行う。このため、ファイルサーバ 140 は、前記 2 つの I/Oパス名を含むノードがローカルノードか否かを判定し、ローカルノードでない場合、すなわちリモートノードである場合、2 つの I/Oパスを含むノード番号に対応するノード（ノード 200、ノード 300）のファイルサーバ 240、340 に I/Oパス名を指定したライト要求を送信する（ステップ 505、506）。

【0116】(4) ステップ 505 での判定が、ノードがローカルノードであった場合、ローカルノードのディスクドライバに I/Oパスを指定したライト要求を送信する（ステップ 507）。

【0117】図 15 に示す例の場合、前述の処理で、ファイルサーバ 140 は、ファイルサーバ 240 に I/Oパス“(200, 290, 11)”を指定したライト要求を送信し、ファイルサーバ 340 に I/Oパス“(300, 390, 22)”を指定したライト要求を送信する。これらのライト要求を受信したファイルサーバ 240、340 は、それぞれのノードのディスクドライバに I/Oパスを指定したライト要求を送信する。

【0118】受信したアクセス要求がリード要求の場合、ファイルサーバ 140 は、前述した I/Oパスのうちで、論理ディスク管理テーブルの最も最初のフィールドに登録されていた I/Oパス“(200, 290, 11)”を使用してアクセスを行う。もし、I/Oパスの障害などの理由により、この I/Oパスを使用してアクセスすることができない場合、順に次のフィールドに登録されている I/Oパスを使用してアクセスを試みる。また、前述の I/Oパスの中で、ローカルノードのノード番号を含むものがあれば、その I/Oパスを最初に使うようにしてもよい。このように、なるべくリモートアクセスを減らすことによって、ネットワークの負荷を減らすことができる。リード処理に使用する I/Oパスが決定した後の処理は、ライト要求の場合と同様である。

【0119】次に、障害発生時、障害が発生した I/Oパスを切り放す処理を説明する。ここでは、ディスクコントローラや I/Oインターフェイスの障害により、ノード 200 に接続されていた物理ディスク装置 20 にアクセスするための I/Oパス“(200, 290, 11)”が

使用不可能になったものとして説明する。

【0120】障害の発生により、ＩＯパス“(200, 290, 11)”が使用できなくなった場合、ノード200のデバイスドライバ250は、このＩＯパスの障害を検出し、障害発生をファイルサーバ240に通知する。

【0121】この通知を受けたファイルサーバ240は、論理ディスク管理テーブル270を更新し、障害発生ＩＯパスの状態フラグを「使用不可」状態にする。ファイルサーバ240は、図5に示したフローによる方法により、全てのリモートノードのファイルサーバと通信を行い、全てのノードの論理ディスク管理テーブルを論理ディスク管理テーブル270と同一の内容に更新する。

【0122】最後に、ファイルサーバ240は、高信頼ディスク装置30が接続されたノード100のファイルサーバ140に、障害発生ＩＯパス“(200, 290, 11)”が「使用不可」状態になったことを、マウント構成ファイル24に記載するように要求する。この要求を受けたファイルサーバは、マウント構成ファイル24を更新し、上記障害発生ＩＯパスの使用可否情報を“unavailable”(使用不可)に書き換える。以上によりＩＯパスの切り離しが終了する。

【0123】ＩＯパスの切り離し処理中に、あるノードのファイルサーバ(例えば、ファイルサーバ140)が、ファイルサーバ240に前述の障害発生ＩＯパスを指定したアクセス要求を送るとその要求は失敗する。しかし、ライト処理の場合、データは、同時に複数の物理ディスク装置に書き込まれるので、アクセス可能な物理ディスク装置(物理ディスク装置20)の方に無事に記録されている。また、リード処理の場合、アクセス要求を行ったファイルサーバは、アクセスに失敗したら別のＩＯパス“(300, 390, 22)”を指定したＩＯアクセス要求をファイルサーバ340に送信する。このため、データは、アクセス可能な物理ディスク装置から無事に読み込まれる。従って、ＩＯパス切り替え中もユーザは、それを意識することなくファイルにアクセスすることができる。

【0124】前述した本発明の実施形態において、ノード200で障害が発生したことにより、ＩＯパス“(200, 290, 11)”が使用できなくなった場合、ノード200にＩＯアクセス要求を送信したリモートノードのファイルサーバが、送信したアクセス要求のタイムアウトによってノード200の障害を検出し、障害を検出したこのファイルサーバが上記のＩＯパスの切り離し処理を行うようにすればよい。

【0125】また、前述した本発明の実施形態において、論理ディスク管理テーブルに、論理ディスクの使用方法(切り替え、ミラーリングなど)を指定するためのディスクタイプ情報を論理ディスクID毎に登録できる

ようにし、マウント構成ファイル24に上記ディスクタイプ情報を登録できるようにし、システム起動時にファイルサーバ140がマウント構成情報24に記載されたディスクタイプ情報を、論理ディスク管理テーブルのディスクタイプ情報に登録し、通常運用時及び障害発生時、ファイルサーバが論理ディスク管理テーブルのディスクタイプ情報によって、ディスクタイプを判別し各ディスクタイプ毎の処理を行うようにすることもできる。例えば、図15に示す例の場合、マウント構成ファイル24には“( (200, 290, 11) available) ( (300, 390, 22) available) /mnt mirror”と記載する。“mirror”は、前述2つのＩＯパスからアクセスされる物理ディスク装置に対して、ミラーリングを行うことを示す。ファイルサーバ140は、起動時に前述のエントリを読み込んで、ディスクタイプが「ミラーリング」であることを判別し、論理ディスク管理テーブルの対応するディスクタイプ情報に、「ミラーリング」であることを登録する。通常運用時、ファイルサーバは、論理ディスク管理テーブルのディスクタイプ情報を参照して、前述のＩＯパスの組が「ミラーリング」を行うものであることを判別すると、前述した実施形態により説明した「ミラーリング」の処理を行う。ディスクタイプが「切り替え」の場合も同様である。これにより、ＩＯパスの切り替えとミラーリングをシステムで共存させることができる。

【0126】前述した本発明の第3の実施形態は、ファイルのミラーリングを行うものとして説明したが、論理ディスク管理テーブルの1つのエントリに登録されたＩＯパスからアクセスされる物理ディスク装置に、ファイルを分散して記録するようにすれば、ファイルのストライピングを行うことができる。

【0127】

【発明の効果】以上説明したように本発明によれば、ＩＯパス切り替え・復旧処理のためにかかる時間を短縮することができ、また、ＩＯパス切り替え時にファイルの整合性のチェックを不要にすることができる。また、本発明によれば、ＩＯパスの切り替え・切り離し処理が発生しても、一般ユーザはそれを意識することなく作業を続けることができる。さらに、本発明によれば、ＩＯパス切り替え・切り離し処理後あるいは障害発生ＩＯパス復旧後、システムを再起動する際にシステム管理者がマウント構成ファイルを設定しなおす必要をなくすることができ、システム管理者の負担を軽減することができる。

【図面の簡単な説明】

【図1】本発明の第1の実施形態によるファイルシステムの構成を示すブロック図である。

【図2】システム内に設けられる各種のテーブルの具体的な構成例を説明する図である。

【図3】マウント構成ファイルの具体的な構成例を説明する図である。

【図 4】 システムの起動時のファイルサーバの処理動作を説明するフローチャートである。

【図 5】 システム全体のノードの論理ディスク管理テーブルを更新する処理動作を説明するフローチャートである。

【図 6】 通常運用時のファイルサーバの処理動作を説明するフローチャートである。

【図 7】 I/Oパスの切り替えの処理動作を説明するフローチャートである。

【図 8】 I/Oパスに障害が発生し I/Oパスの切り替えを行う処理について説明する図（その 1）である。

【図 9】 I/Oパスに障害が発生し I/Oパスの切り替えを行う処理について説明する図（その 2）である。

【図 10】 I/Oパスに障害が発生し I/Oパスの切り替えを行う処理について説明する図（その 3）である。

【図 11】 I/Oパスが障害から復旧したとき、I/Oパスをシステムに復旧させる処理手順を説明するフローチャートである。

【図 12】 I/Oパス切り替え時の障害発生ノードの処理動作の他の例について説明するフローチャートである。

【図 13】 障害発生ノード以外のノードの処理動作の他の例を説明するフローチャートである。

【図 14】 本発明の第 2 の実施形態によるディスクキャ

ッシュの書き戻しの処理とバッファキャッシュの転送の処理とを説明する図である。

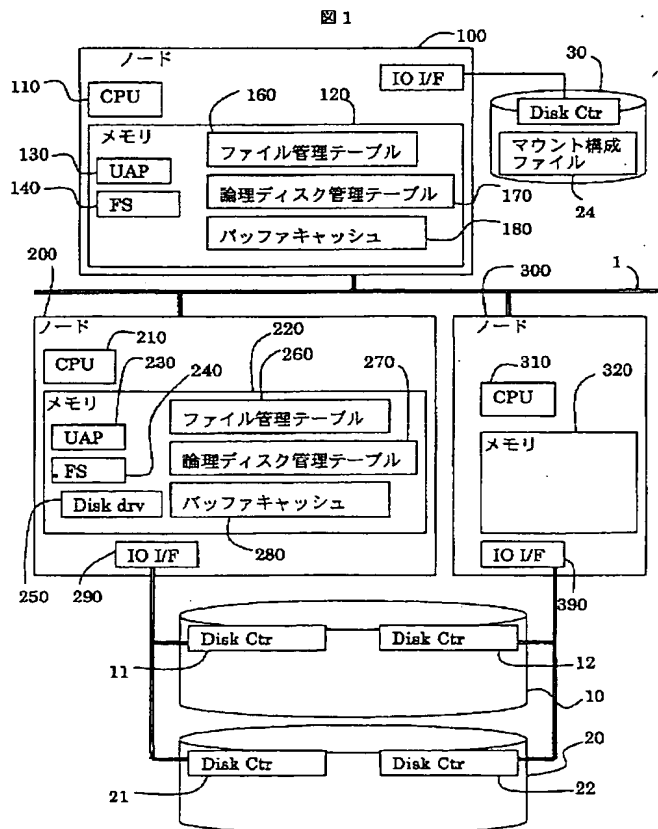
【図 15】 本発明の第 3 の実施形態によるファイルシステムの構成を示すブロック図である。

【図 16】 本発明の第 3 の実施形態におけるマウント構成ファイルの具体的な構成例を説明する図である。

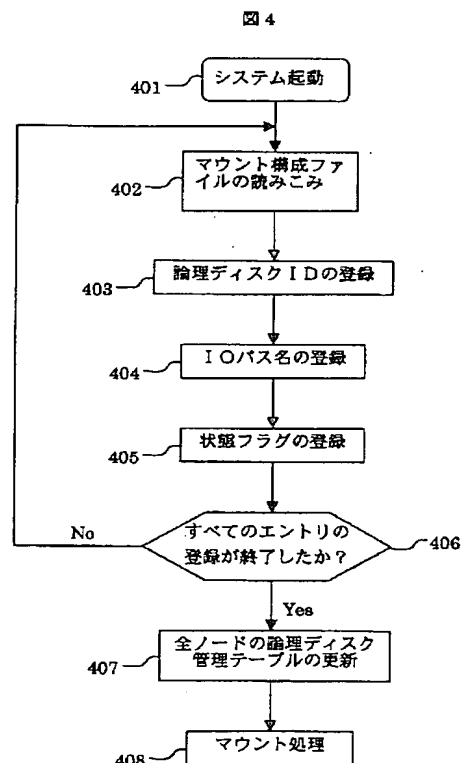
【符号の説明】

- 1 ネットワーク
- 10、20、30 物理ディスク装置
- 11、12、21、22 ディスクコントローラ
- 13 ディスクキャッシュ
- 24 マウント構成ファイル
- 100、200、300 ノード
- 110、210、310 CPU
- 120、220、320 メモリ
- 130、230 ユーザアプリケーション（UAP）
- 140、240、340 ファイルサーバ（FS）
- 160、260 ファイル管理テーブル
- 170、270 論理ディスク管理テーブル
- 180、280、360 バッファキャッシュ
- 250、350 ディスクドライバ
- 290、390 I/Oインタフェース

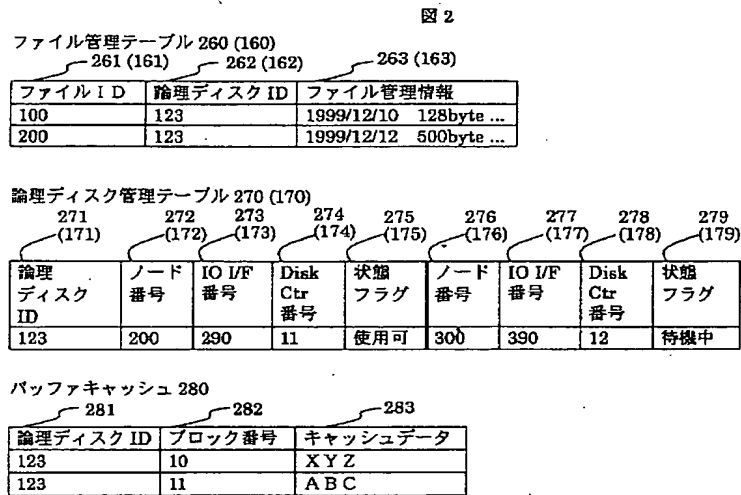
【図 1】



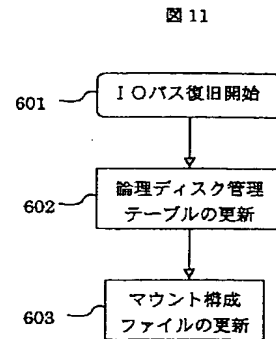
【図 4】



【図 2】



【図 11】



【図 3】

図 3

マウント構成ファイル 24

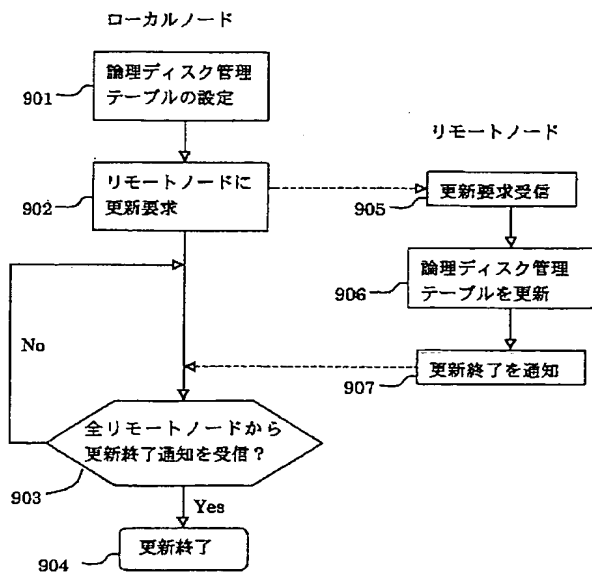
# ((ノード番号, IO I/F 番号, Disk Ctr 番号) 使用可否) マウントポイント ← コメント行

((200, 290, 11) available) ((300, 390, 12) available) /mnt

51 52 53 54 55

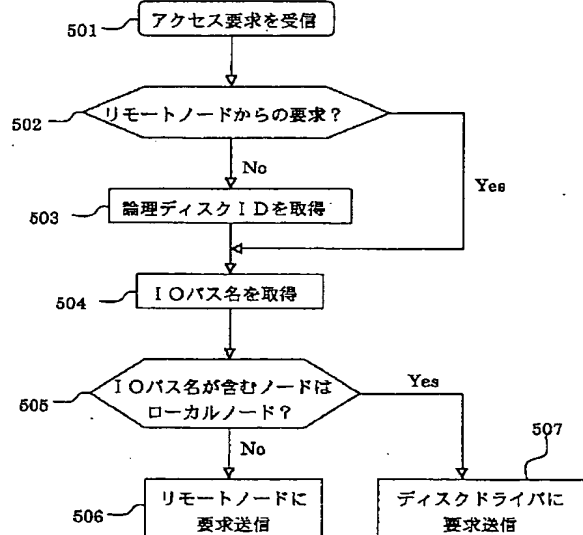
【図 5】

図 5

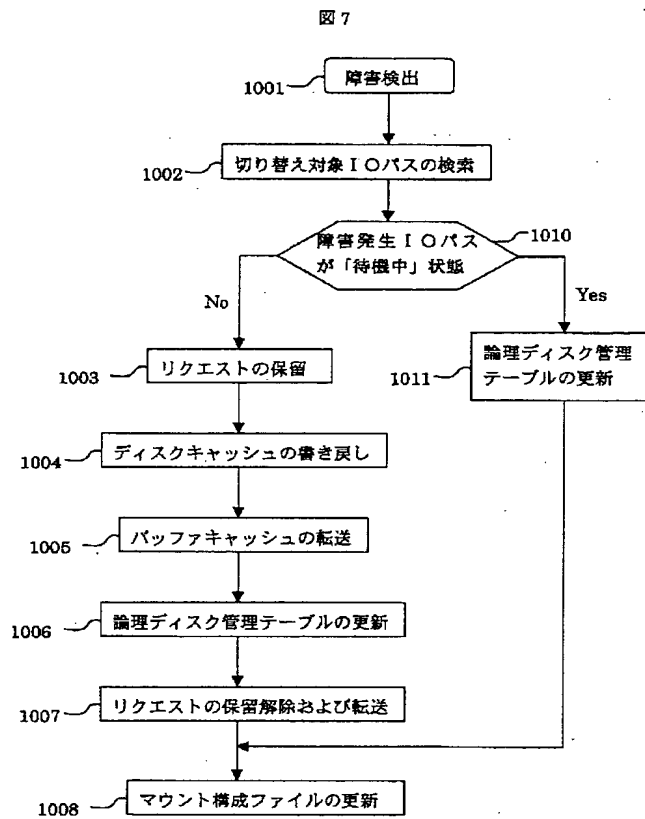


【図 6】

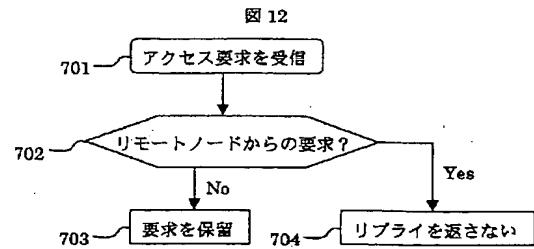
図 6



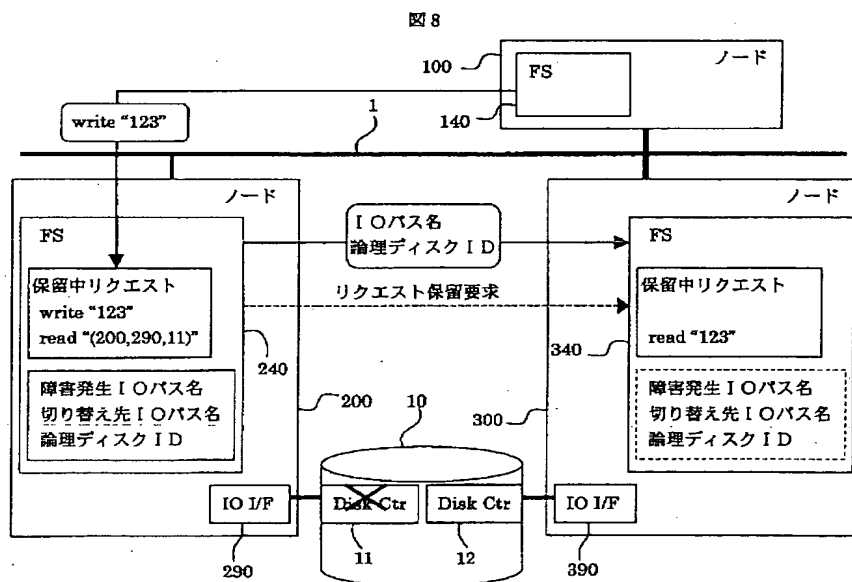
【図7】



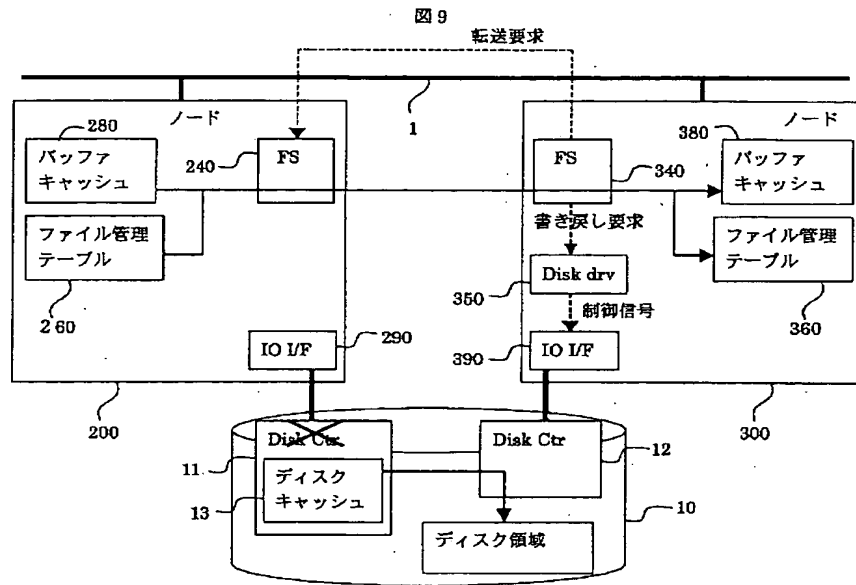
【図12】



【図8】

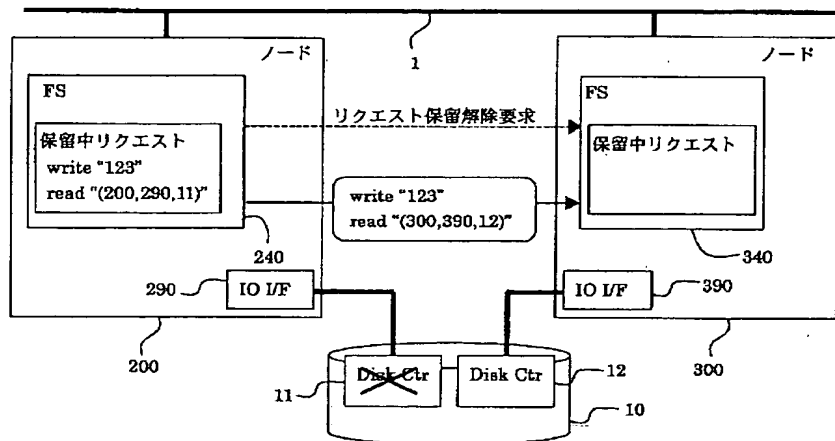


【図9】



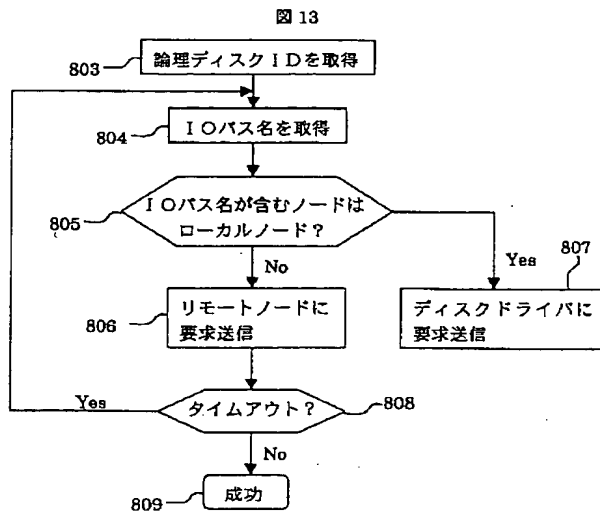
【図10】

図10



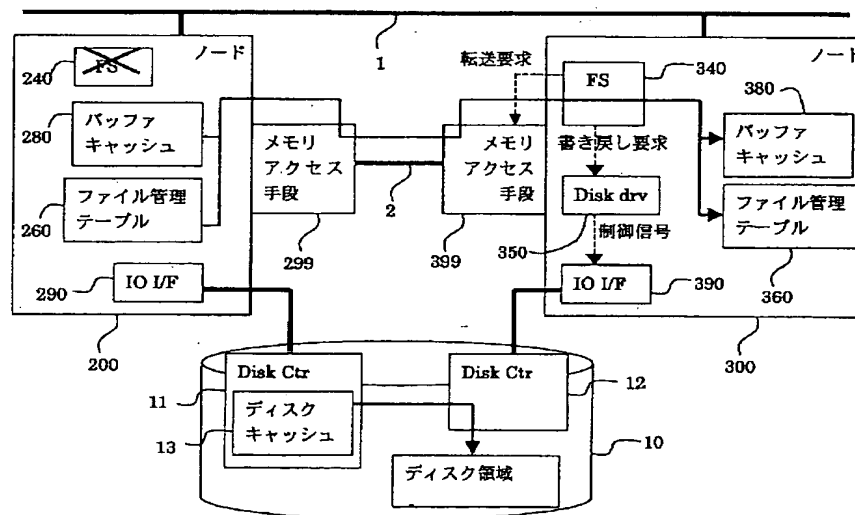


【図13】



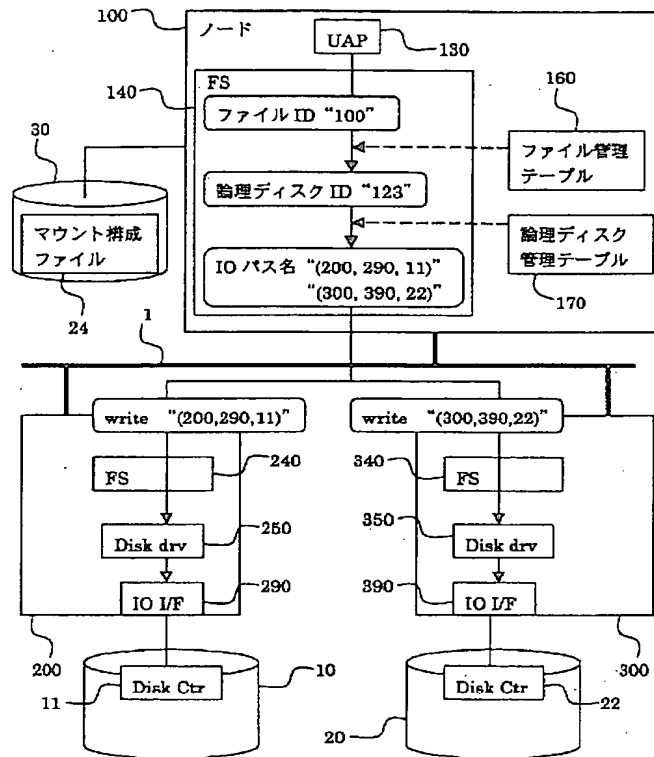
【図14】

図14



【図15】

図15



【図16】

図16

マウント構成ファイル 24

# ((ノード番号, IO I/F 番号, Disk Ctr 番号) 使用可否) マウントポイント ← コメント行

((200, 290, 11) available) ((300, 390, 22) available) /mnt

51 52 53 54 55

論理ディスク管理テーブル 270 (170)

論理 ディスク ID	ノード 番号	IO I/F 番号	Disk Ctr 番号	状態 フラグ	ノード 番号	IO I/F 番号	Disk Ctr 番号	状態 フラグ
123	200	290	11	使用可	300	390	22	使用可

フロントページの続き

(72)発明者 藺田 浩二

神奈川県川崎市麻生区王禅寺1099番地 株  
式会社日立製作所システム開発研究所内

(72)発明者 熊△崎▽ 裕之

神奈川県横浜市戸塚区戸塚町5030番地 株  
式会社日立製作所ソフトウェア事業部内

Fターム(参考) 5B014 HA09 HA13 HB01 HB26  
5B018 GA10 HA40 KA11 MA12 QA01  
5B065 CC01 EA12  
5B082 EA01 FA05  
5B083 AA08 BB03 CC04 CD11 EE08